Dokumentacja projektu zaliczeniowego

Przedmiot: Inżnierski projekt zespołowy 2

Temat: **Aplikacja mobilna do zarządzania listą zadań domowych.**

Autorzy: **Monika Rozmarynowska**

**Jakub Kucharski**

**Krzysztof Bieniek**

**Olaf Maliszewski**

**Krzysztof Kubiś**

**Piotr Wojdalski**

Grupa: 320

Kierunek: informatyka

Rok akademicki: 3

Poziom i semestr: I/6

Tryb studiów: stacjonarne

Należy pozostawić wszelkie nagłówki tego dokumentu, a umieszczać treść w odpowiednich miejscach zamiast obecnych objaśnień.

Stronę tytułową można sformatować w dowolny sposób, ale należy pozostawić zawartość informacyjną w układzie pokazanym powyżej.

Praca powinna zostać złożona wyłącznie w formacie pdf. Przed wygenerowaniem ostatecznej wersji należy zaktualizować spis treści – wyświetlane dwa poziomy.

Niniejszą informację należy również usunąć z wersji końcowej.

1. Spis treści

[2 Odnośniki do innych źródeł 4](#_Toc1976793)

[3 Słownik pojęć 5](#_Toc1976794)

[4 Wprowadzenie 6](#_Toc1976795)

[4.1 Cel dokumentacji 6](#_Toc1976796)

[4.2 Przeznaczenie dokumentacji 6](#_Toc1976797)

[4.3 Opis organizacji lub analiza rynku 6](#_Toc1976798)

[4.4 Analiza SWOT organizacji 6](#_Toc1976799)

[5 Specyfikacja wymagań 7](#_Toc1976800)

[5.1 Charakterystyka ogólna 7](#_Toc1976801)

[5.2 Wymagania funkcjonalne 7](#_Toc1976802)

[5.3 Wymagania niefunkcjonalne 8](#_Toc1976803)

[6 Zarządzanie projektem 9](#_Toc1976804)

[6.1 Zasoby ludzkie 9](#_Toc1976805)

[6.2 Harmonogram prac 9](#_Toc1976806)

[6.3 Etapy/kamienie milowe projektu 9](#_Toc1976807)

[7 Zarządzanie ryzykiem 10](#_Toc1976808)

[7.1 Lista czynników ryzyka 10](#_Toc1976809)

[7.2 Ocena ryzyka 10](#_Toc1976810)

[7.3 Plan reakcji na ryzyko 10](#_Toc1976811)

[8 Zarządzanie jakością 11](#_Toc1976812)

[8.1 Scenariusze i przypadki testowe 11](#_Toc1976813)

[9 Projekt techniczny 12](#_Toc1976814)

[9.1 Opis architektury systemu 12](#_Toc1976815)

[9.2 Technologie implementacji systemu 12](#_Toc1976816)

[9.3 Diagramy UML 12](#_Toc1976817)

[9.4 Charakterystyka zastosowanych wzorców projektowych 12](#_Toc1976818)

[9.5 Projekt bazy danych 12](#_Toc1976819)

[9.6 Projekt interfejsu użytkownika 12](#_Toc1976820)

[9.7 Procedura wdrożenia 13](#_Toc1976821)

[10 Dokumentacja dla użytkownika 14](#_Toc1976822)

[11 Podsumowanie 15](#_Toc1976823)

[11.1 Szczegółowe nakłady projektowe członków zespołu 15](#_Toc1976824)

[12 Inne informacje 16](#_Toc1976825)

1. Odnośniki do innych źródeł
   * Zarządzania projektem – [Trello](https://trello.com/b/j6LRgglz/chronos)
   * Repozytorium kodu – [GitHub](https://github.com/Rozmarynka19/Chronos)
2. Słownik pojęć

Tabela lub lista z pojęciami, które wymagają wyjaśnienia, wraz z tymi wyjaśnieniami – w szczególności synonimy różnych pojęć używanych w dokumentacji.

1. Użytownik – osoba korzystająca z aplikacji
2. Wprowadzenie
   1. Cel dokumentacji

po co ją robimy i co zawiera (poziom szczegółowości)

* 1. Przeznaczenie dokumentacji

dla kogo ona jest

* 1. Opis organizacji lub analiza rynku

Jedna z dwóch opcji:

1. Jeśli dla konkretnej organizacji: Czym jest organizacja, dla której realizowany będzie system; jak działa (lub będzie działała), kiedy system będzie wdrożony – tutaj nie odwołujemy się do samego systemu, tylko opisujemy samo działanie organizacji i role. W szczególności – jak wyglądają główne procesy biznesowe.
2. Jeśli na masowy rynek: Pobieżna analiza rynku. Dla kogo będzie przydatny taki system. Ile jest organizacji, które będą mogły z niego skorzystać, użytkowników w poszczególnych organizacjach. Czy te organizacje stanowią jednorodną grupę czy są różne rodzaje. Co one mają ze sobą wspólnego. Jak ta liczba będzie się zmieniała w najbliższej przyszłości.
   1. Analiza SWOT organizacji

Tylko jeśli dla konkretnej organizacji

Wystarczy sama tabela 2x2 (silne-słabe-szanse-zagrożenia)

1. Specyfikacja wymagań
   1. Charakterystyka ogólna
      1. Definicja produktu

System do nauki i ćwiczeń algorytmów, prezentujący zagadnienia teoretyczne oraz praktyczne z dziedziny algorytmiki.

* + 1. Podstawowe założenia

System ma na celu ułatwienie oraz przyśpieszenie nauki algorytmów studentom kierunków technicznych, umożliwienie poznania problemu z perspektywy innych studentów, zapewnienie dostępu do zakresu materiału, zgrupowanego w jednym miejscu. System może służyć również jako narzędzie dla dydaktyków w ramach zajęć.

* + 1. Cel biznesowy

Usprawnienie nauczania, poprawa wyników oraz efektywności w nauce, zapewnienie dodatkowego źródła wiedzy z danego tematu.

* + 1. Użytkownicy

1. Użytkownik
   * 1. Korzyści z systemu
2. Użytkownik
   1. Dostęp do materiałów naukowych.
   2. Możliwość rozwiązywania testów w celu nauki.
   3. Dostęp do programistycznej realizacji algorytmów.
   4. Dostęp do wizualizacji problemów algorytmicznych.
   5. Możliwość monitorowania swojego postępu w nauce.
      1. Ograniczenia projektowe i wdrożeniowe

System do poprawnego działania potrzebuje komputera z systemem operacyjnym z rodziny Windows.

* 1. Wymagania funkcjonalne
     1. Lista wymagań

lista numerowana – czyli lista przypadków użycia lub bardziej ogólnie sformułowane wymagania

1. Użytkownik ma możliwość dodawania zadań do listy
2. Użytkownik ma możliwość tworzenia osobnych list – stan początkowy – parę list, np. „obowiązki domowe”, „auto”, itd.
3. Użytkownik ma możliwość tworzenia dodatkowych list.
4. Użytkownik ma możliwość podglądu zadań w kalendarzu wbudowaną w aplikację.
5. Użytkownik ma możliwość dodawania w zadaniu terminu wykonania zadań.
6. Użytkownik ma możliwość dodania opisu do zadania.
7. Użytkownik ma możliwość dodania cykliczności zadania.
8. Użytkownik ma możliwość dodawania załączników do danego zadania
9. Użytkownik ma możliwość dodania podlisty zadań w zadaniu
10. Użytkownik ma możliwość ustawienia przypomnień wykonania zadania (push-notification)
11. Użytkownik ma możliwość oznaczania piorytetu danego zadania
12. Użytkownik ma możliwość ustawienia dźwięku dla przypomnień [personalizacja]
13. Użytkownik ma możliwość podglądu zadań do zrobienia dzisiaj i których termin jest ustawiony na jutro (typu „Hej, Marku! Masz tu do zrobienia dzisiaj, to i tamto i siamto [lista zadań]. A na jutro masz: bla bla bla bla [lista zadań])
14. Użytkownik ma możliwość zalogowania się do aplikacji za pomocą loginu i hasła. Przy rejestracji podaje login, hasło, email, numer telefonu. Następny etap logowania polega na wprowadzniu kodu, który zostałby wyslany na maila/SMSem na podany przy rejestracji numer telefonu.
15. Użytkownik ma możliwość zalogowania się do aplikacji za pomocą konta na Google’u
16. Użytkownik ma możliwość zeskanowanie kodu QR faktury i dopisuje do listy rachunków. Zapisuje sobie fakturę na przyszłość.
17. Użytkownik ma możliwość zrealizowania płatności zapisanej faktury. [dodatkowa funkcjonalność = bramka płatności; do zrealizowania w momencie opracowania podstawowych funkcjonalności]
18. Użytkownik ma możliwość posortować zadania względem priorytetu zadania.
19. Użytkownik ma możliwość posortować zadania względem terminu.
20. Użytkownik ma możliwość posortować zadania względem daty utworzenia.
21. Użytkownik ma możliwość ręcznego wprowadzenia danych do przelewu i zapisania go na przyszłość.
22. Użytkownik ma możliwość zeskanowania dokumentu i rozpoznania odpowiednich pól z faktury (OCR)
23. Użytkownik ma możliwość zapisania swoich danych do przelewu i udostępnianie innym użytkownikom naszej aplikacji
24. Aplikacja musi synchronizować obecny stan konta użytkownika (listy zadań, same zadania, załączniki) między urządzeniami, tj. po zalogowaniu się odpowiednimi danymi na obcym urządzeniu użytkownik ma dostęp do wszystkich swoich zadań z własnego urządzenia
25. 6 Zarządzanie projektem
26. 6.1 Zasoby ludzkie
27. Programiści (3 osoby)
28. Tester (1 osoba)
29. UI designer (1 osoba)
30. Kierownik projektu (1 osoba)

6.2 Harmonogram prac

6.3 Etapy/kamienie milowe projektu

1. Opracowanie specyfikacji wymagań
   1. wymagania funkcjonalne
   2. wymagania niefunkcjonalne
2. Ocena ryzyka projektu
3. Zaprojektowanie architektury aplikacji
   1. diagramy przypadków użycia
   2. diagramy klas
   3. diagramy komponentów
   4. diagramy pakietów
   5. diagramy maszyny stanowej
   6. diagramy komunikacji
4. Zaprojektowanie GUI
5. Opracowanie testów
   1. opracowanie testów jednostkowych
   2. opracowanie testów integracyjnych
   3. opracowanie testów akceptacyjnych
6. Wprowadzenie poprawek do architektury aplikacji
7. Wytworzenie pakietów
8. Przeprowadzenie testów jednostkowych
9. Integracja pakietów
10. Przeprowadzenie testów integracyjnych
11. Wdrożenie systemu
12. Testowanie systemu w środowisku pracy
    * 1. Diagramy przypadków użycia

Tutaj same diagramy – bez specyfikacji, ale każdy diagram z tytułem i na osobnej stronie

* + 1. Szczegółowy opis wymagań

każde na nowej stronie wg następujących punktów:

* Numer – jako ID
* Nazwa
* Uzasadnienie biznesowe – odwołanie (-a) do elementów wymienionych w 5.1.5. (id i treść elementu, do którego się odwołujemy)
* Użytkownicy
* Scenariusze, dla każdego z nich:
* Warunki początkowe
* **Przebieg działań –** numerowana lista kroków, ze wskazaniem, kto realizuje dany krok
* Efekty – warunki końcowe
* Wymagania niefunkcjonalne – szczegółowe wobec poszczególnych wymagań funkcjonalnych
* Częstotliwość - na skali 1-5 lub BN-BW
* Istotność – inaczej: zależność krytyczna, znaczenie - na skali 1-5 lub BN-BW

***Ważne!***

*Elementy od warunków początkowych do końca mogą być grupowane, tj. specyfikacja pojedynczego przypadku użycia może zawierać:*

*- pojedynczy przebieg działań (scenariusz główny) oraz ew. scenariusze alternatywne, albo*

*- wiele przebiegów głównych wraz z ew. scenariuszami alternatywnymi – wtedy każdy z przebiegów głównych powinien być opisany wg tych punktów (od warunków początkowych do końca).*

* 1. Wymagania niefunkcjonalne

1. Wydajność
   1. Wyświetlenie informacji o zadaniu, po wybraniu go z listy, nie powinno trwać dłużej niż 0.5s.
   2. Aplikacja powinna zalogować (po wcześniejszej autoryzacji) użytkownika po uruchomieniu w ciągu 1 sekundy.
   3. Dołączenie załącznika do zadania ważącego nie więcej niż 5MB powinno zająć nie więcej niż 1 sekundę.
   4. Sortowanie zadań nie powinno zająć więcej niż 0.1 sekundę.
   5. Zeskanowanie kodu QR i wprowadzenie danych do formularza w aplikacji nie powinno zająć więcej niż 2 sekundy.
   6. Zeskanowanie faktury i wprowadzenie danych do formularza w aplikacji nie powinno zająć więcej niż 2 sekundy.
   7. Kod uwierzytelniający powinien dojść w ciągu 1 minuty.
2. Inne cechy jakości
   1. System powinien działać na smartfonach z systemem Android 4.1 (Jelly Bean, API 16) w górę.
3. Zarządzanie projektem
   1. Zasoby ludzkie

(rzeczywiste lub hipotetyczne) – przy realizacji projektu

Należy założyć, że projekt byłby realizowany w całości jako projekt komercyjny a nie tylko częściowo w ramach zajęć na uczelni

* 1. Harmonogram prac

Etapy mogą się składać z zadań.

Wskazać czasy trwania poszczególnych etapów i zadań – wykres Gantta.

obejmuje również harmonogram wdrożenia projektu – np. szkolenie, rozruch, konfiguracja, serwis – może obejmować różne wydania (tj. o różnej funkcjonalności – personal, professional, enterprise) i wersje (1.0, 1.5, itd.)

* 1. Etapy/kamienie milowe projektu

dla głównych etapów projektu

1. Zarządzanie ryzykiem
   1. Lista czynników ryzyka

Wypełniona lista kontrolna

* 1. Ocena ryzyka

prawdopodobieństwo i wpływ

* 1. Plan reakcji na ryzyko

Działania w odniesieniu do poszczególnych ryzyk.

Mogą być wg różnych strategii, tj. kilka strategii dla pojedynczego czynnika ryzyka

1. Zarządzanie jakością
   1. Scenariusze i przypadki testowe

szczegółowy plan testowania systemu – głównie testowanie funkcjonalności; każdy scenariusz od nowej strony, musi zawierać co najmniej następujące informacje (sugerowany układ tabelaryczny, np. wg szablonu podanego w osobnym pliku lub na wykładzie):

* numer – jako ID
* nazwa scenariusza – co test w nim testowane (max kilka wyrazów)
* kategoria – poziom/kategoria testów
* opis – dodatkowe opcjonalne informacje, które nie zmieściły się w nazwie
* tester - konkretna osoba lub klient/pracownik,
* termin – kiedy testowanie ma być przeprowadzane,
* narzędzia wspomagające – jeśli jakieś są używane przy danym scenariuszu
* przebieg działań – tabela z trzema kolumnami: lp. oraz opisującymi działania testera i systemu
* założenia, środowisko, warunki wstępne, dane wejściowe – przygotowanie przed uruchomieniem testów
* zestaw danych testowych – najlepiej w formie tabelarycznej – jakie konkretnie dane mają być użyte przez testera i zwrócone przez system w poszczególnych krokach przebiegu działań
* *przebieg lub zestaw danych testowych musi zawierać jawną informację o warunku zaliczenia testu*

1. Projekt techniczny
   1. Opis architektury systemu

z ew. rysunkami pomocniczymi

* 1. Technologie implementacji systemu

tabela z listą wykorzystanych technologii, każda z uzasadnieniem

* 1. Diagramy UML

każdy diagram ma mieć tytuł oraz ma być na osobnej stronie

diagramy przypadków użycia umieszczone w punkcie 1, a nie tutaj.

* + 1. Diagram(-y) klas
    2. Diagram(-y) czynności
    3. Diagramy sekwencji

co najmniej 5, w tym co najmniej 1 przypadek użycia zilustrowany kilkoma diagramami sekwencji

* + 1. Inne diagramy

co najmniej trzy – komponentów, rozmieszczenia, maszyny stanowej itp.

* 1. Charakterystyka zastosowanych wzorców projektowych

informacja opisowa wspomagana diagramami (odsyłaczami do diagramów UML); jeśli wykorzystano wzorce projektowe, to należy wykazać dwa z nich

* 1. Projekt bazy danych
     1. Schemat

w trzeciej formie normalnej; jeśli w innej to umieć uzasadnić wybór

* + 1. Projekty szczegółowe tabel
  1. Projekt interfejsu użytkownika

Co najmniej dla głównej funkcjonalności programu – w razie wątpliwości, uzgodnić z prowadzącym zajęcia

* + 1. Lista głównych elementów interfejsu

okien, stron, aktywności (Android)

* + 1. Przejścia między głównymi elementami
    2. Projekty szczegółowe poszczególnych elementów

każdy element od nowej strony z następującą minimalną zawartością:

* numer – ID elementu
* nazwa – np. formularz danych produktu
* projekt graficzny – wystarczy schemat w narzędziu graficznym lub zrzut ekranu – z przykładowymi informacjami (nie pusty!!!)
* opcjonalnie:
* opis – dodatkowe opcjonalne informacje o przeznaczeniu, obsłudze – jeśli nazwa nie będzie wystarczająco czytelna
* wykorzystane dane – jakie dane z bazy danych są wykorzystywane
* opis działania – tabela pokazująca m.in. co się dzieje po kliknięciu przycisku, wybraniu opcji z menu itp.
  1. Procedura wdrożenia

jeśli informacje w harmonogramie nie są wystarczające (a zapewne nie są)

1. Dokumentacja dla użytkownika

Opcjonalnie – dla chętnych

Na podstawie projektu docelowej aplikacji, a nie zaimplementowanego prototypu architektury

4-6 stron z obrazkami (np. zrzuty ekranowe, polecenia do wpisania na konsoli, itp.)

* pisana językiem odpowiednim do grupy odbiorców – czyli najczęściej nie do informatyków
* może to być przebieg krok po kroku obsługi jednej głównej funkcji systemu, kilku mniejszych, instrukcja instalacji lub innej pomocniczej czynności.

1. Podsumowanie
   1. Szczegółowe nakłady projektowe członków zespołu

tabela (kolumny to osoby, wiersze to działania) pokazująca, kto ile czasu poświęcił na projekt oraz procentowy udział każdej osoby w danym zadaniu oraz wiersz podsumowania – udział każdej osoby w skali całego projektu

1. Inne informacje

przydatne informacje, które nie zostały ujęte we wcześniejszych punktach