



WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI
I INFORMATYKI

Dokumentacja Projektu grupowego

Harmonogram i specyfikacja wymagań

Wydział Elektroniki, Telekomunikacji i Informatyki

Politechnika Gdańska

Nazwa i akronim projektu: {nazwa projektu, np: System zabezpieczenia portu przed zagrożeniami terrorystycznymi - SZP} Aplikacja wizualizująca zagadnienia związane z kodowaniem nadmiarowym	Zleceniodawca: {nazwa/nazwisko klienta} dr inż. Bartosz Czaplewski	
Numer zlecenia: {numer zespołu projektowego w ramach Projektu grupowego wg systemu SPG, np. 13@KSSR'2022} 5@KSTI'2023/24	Kierownik projektu: {kierownik zespołu projektowego} Bartosz Kołakowski	Opiekun projektu: {opiekun projektu} dr inż. Bartosz Czaplewski

Nazwa / kod dokumentu: Harmonogram i specyfikacja wymagań – HiSW	Nr wersji: {wersja dokumentu np. 1.00} 6.00
Odpowiedzialny za dokument: {nazwisko, imię} Kołakowski Bartosz	Data pierwszego sporządzenia: {data wykonania pierwszej wersji dokumentu} 31.10.2023
	Data ostatniej aktualizacji: {data wykonania aktualnej wersji dokumentu} 28.05.2024
	Semestr realizacji Projektu grupowego: {wpisać 1 lub 2} 2

Historia dokumentu

Wersja	Opis modyfikacji	Rozdział / strona	Autor modyfikacji	Data
1.00	{opis np. wstępna wersja} Wstępna wersja	{np. całość} całość	{nazwisko, imię} Kołakowski Bartosz	{data zmiany} 31.10.2023
2.00	{opis np. wstępna wersja} Dodanie dat do harmonogramu prac	{np. całość} pkt 2	{nazwisko, imię} Kołakowski Bartosz	{data zmiany} 8.11.2023
3.00	{opis np. wstępna wersja} Bardziej szczegółowe opisy etapów projektu	{np. pkt 2, 2.3} pkt 2.1	{nazwisko, imię} Kołakowski Bartosz	{data zmiany} 21.11.2023
4.00	{opis np. wstępna wersja} Zmiana dat w harmonogramie, uszczegółowienie punktu trzeciego	{np. pkt 2, 2.3} pkt 2 i 3	{nazwisko, imię} Jastrzębski Paweł	{data zmiany} 01.01.2024
5.00	{opis np. wstępna wersja} Dopisanie informacji w punkcie trzecim	{np. pkt 2, 2.3} pkt 3	{nazwisko, imię} Kołakowski Bartosz	{data zmiany} 04.01.2024
6.00	{opis np. wstępna wersja} Aktualizacja informacji	{np. pkt 2, 2.3} całość	{nazwisko, imię} Kołakowski Bartosz	{data zmiany} 28.05.2024
{wersja}	{opis np. dodanie etapu C}	{np. pkt 2, 2.3}	{nazwisko, imię}	{data zmiany}

{UWAGA: w II semestrze dokumentacja może być rozszerzeniem dokumentacji z semestru I (nowa wersja dokumentu), może być też nowym plikiem

UWAGA: harmonogram może być zaplanowany w I semestrze od razu na 2 semestry – należy jednak w drugim semestrze zaktualizować plik z I semestru; może on również zawierać dodatkowe postanowienia, ustalone po zakończeniu I semestru

UWAGA: harmonogram utworzony w systemie SPG nie musi być taki sam, jak w niniejszym dokumencie}

Spis treści

1	Wprowadzenie - o dokumencie.....	4
1.1	Cel dokumentu	4
1.2	Odbiorcy	4
1.3	Terminologia	4
2	Harmonogram prac zespołu projektowego	4
2.1	Opis etapów wytwarzania (prowadzenia projektu).....	4
	stworzenie bazy projektu.....	4
	stworzenie bazy GUI	5
	implementacja kodu Hamminga	5
	implementacja kodu Reeda-Solomona	5
	wizualizacja kodu Hamminga	5
	wizualizacja kodu Reeda-Solomona	5
	napisanie testów do kodu Hamminga	5
	napisanie testów do kodu Reeda-Solomona	5
	napisanie tłumaczenia programu na język angielski	5
3	Planowany podział zadań i ról w projekcie w zespole projektowym	6
3.1	Opis zadań planowanych do realizacji ze wskazaniem osób odpowiedzialnych	6
	Stworzenie repozytorium na GitHub	6
	Code Review	6
	Projektowanie interfejsu użytkownika	6
	Pisanie testów	6
	Zarządzenie zespołem	6
	Implementacja kodu nadmiarowego	6
	Tworzenie wizualizacji kodu nadmiarowego	6
	Napisanie tłumaczenia na język angielski	6
	Pisanie dokumentów	6
	Stworzenie prezentacji	6
4	Wymagania dla produktu i kryteria akceptacji	7
4.1	Ogólny opis planowanego produktu	7
4.2	Wymagania minimalne dla produktu	7
4.3	Warunki odbioru	7
5	Postanowienia	7
5.1	Postanowienia w zakresie zmian w stosunku do pierwotnego planu i zakresu prac	7
5.2	Inne postanowienia	7

1 Wprowadzenie - o dokumencie

1.1 Cel dokumentu

{nie zmieniać}

Celem dokumentu udokumentowanie zaplanowanego harmonogramu realizacji projektu w semestrze, planowanego podziału zadań w zespole projektowym, wskazanie i opisanie zadań oraz ról osób odpowiedzialnych, a także wyspecyfikowanie wymagań dla projektu wraz z kryteriami akceptacji, nałożonych przez opiekuna i klienta.

1.2 Odbiorcy

{określenie adresatów dokumentu, może być to typ odbiorcy; tu: zleceniobiorca (Katedra), członkowie zespołu projektowego oraz wymienione z nazwiska osoby, do których dokument ma dotrzeć}

Zleceniodawca - dr inż. Bartosz Czaplewski (KSTI)

Członkowie zespołu:

Bartosz Kołakowski

Michał Mróz

Paweł Jastrzębski

Maksym Nowak

Piotr Noga

1.3 Terminologia

{wyjaśnienie używanych w dokumencie pojęć i skrótów, oznaczenia używane wewnątrz dokumentu np. oznaczenia wymagań}

Skróty osób:

Bartosz Kołakowski - BK

Michał Mróz - MM

Paweł Jastrzębski - PJ

Maksym Nowak - MN

Piotr Noga – PN

GUI – Graphical User Interface – graficzny interfejs użytkownika

2 Harmonogram prac zespołu projektowego

{główne etapy, wykonawcy, początek, koniec - w formie tabeli; należy zdefiniować przynajmniej 3 główne etapy realizacji projektu w semestrze}

Stworzenie bazy projektu, implementacja kodu Hamminga, implementacja kodu Reda-Salomona, tworzenie bazy GUI, wizualizacja kodu Hamminga, wizualizacja kodu Reda-Salomona, napisanie testów do kodu Hamminga, napisanie testów do kodu R-S

Nazwa	BK	MM	PJ	MN	PN	Rozpoczęcie	Zakończenie
stworzenie bazy projektu		X				31.10.2023	5.11.2023
stworzenie bazy GUI		X			X	31.10.2023	15.11.2023
implementacja kodu Hamminga		X	X			03.11.2023	01.12.2024
implementacja kodu Reeda-Solomona			X	X		21.02.2024	30.04.2024
wizualizacja kodu Hamminga	X	X		X	X	05.11.2023	10.01.2024
wizualizacja kodu Reeda-Solomona		X	X	X	X	01.03.2024	31.05.2024
napisanie testów do kodu Hamminga	X	X				05.11.2023	22.11.2024
napisanie testów do kodu Reeda-Solomona	X		X	X		01.03.2024	31.05.2024
napisanie tłumaczenia na język angielski					X	01.04.2024	31.05.2024

2.1 Opis etapów wytwarzania (prowadzenia projektu)

{główne zadania w poszczególnych etapach}

stworzenie bazy projektu

{cele, produkty, kryteria akceptacji, główne zadania itp.}

Cele: Wybranie, technologii wstępne przygotowanie do użycia tej technologii, stworzenie projektu na Githubie

Produkty: repozytorium na githubie, do którego dostęp ma każdy członek grupy, do repozytorium wrzucona bazowa konfiguracja frameworka QT (c++)

Kryteria akceptacji: dostęp członków grupy do repozytorium z uprawnieniami do pusha (oprócz na branch main) i możliwość odpalenia projektu

stworzenie bazy GUI

{cele, produkty, kryteria akceptacji, główne zadania itp.}

Cele: Bazowy interfejs użytkownika, stworzenie opcji wyboru wizualizacji kodu, opcje wstawienia kodu

Produkty: kod w QT prezentujący GUI, w którym można wybrać, jaki kod chce się zwizualizować (jeszcze bez implementacji tego etapu)

Kryteria akceptacji: estetyczne GUI, odpala się na komputerze każdego członka grupy, dostępne na Linuxie i Windowsie

implementacja kodu Hamminga

{cele, produkty, kryteria akceptacji, główne zadania itp.}

Cele: Stworzenia algorytmu kodu Hamminga

Produkty: kod w c++ pozwalający na zakodowanie danych kodem Hamminga, funkcja do zakodowania informacji oraz funkcja do zdekodowania

Kryteria akceptacji: algorytm poprawnie zaimplementowany, zgodny z teorią kodowania

implementacja kodu Reeda-Solomona

{cele, produkty, kryteria akceptacji, główne zadania itp.}

Cele: Stworzenia algorytmu kodu Reeda-Solomona

Produkty: kod w c++ pozwalający na zakodowanie danych kodem Reeda-Solomona, funkcja do zakodowania informacji oraz funkcja do zdekodowania

Kryteria akceptacji: algorytm poprawnie zaimplementowany, zgodny z teorią kodowania

wizualizacja kodu Hamminga

{cele, produkty, kryteria akceptacji, główne zadania itp.}

Cele: Stworzenie wizualizacji kodu Hamminga z możliwością inputu użytkownika

Produkty: kod w c++ (framework QT) pozwalający na wyświetlenie działania kodu Hamminga dla wprowadzonego inputu

Kryteria akceptacji: wizualizacja w GUI w sposób zrozumiały i estetyczny pokazuje działanie kodu Hamminga, poprawne przyjmowanie inputu użytkownika. Najpierw koduje wprowadzony ciąg bitów, następnie pozwala na zmianę wyniku kodowania, a następnie przeprowadza dekodowanie

wizualizacja kodu Reeda-Solomona

{cele, produkty, kryteria akceptacji, główne zadania itp.}

Cele: Stworzenie wizualizacji kodu Reeda-Solomona z możliwością inputu użytkownika

Produkty: kod w c++ (framework QT) pozwalający na wyświetlenie działania kodu Reeda-Solomona dla wprowadzonego inputu

Kryteria akceptacji: wizualizacja w GUI w sposób zrozumiały i estetyczny pokazuje działanie kodu Reeda-Solomona, poprawne przyjmowanie inputu użytkownika. Najpierw koduje wprowadzony ciąg bitów, następnie pozwala na zmianę wyniku kodowania, a następnie przeprowadza dekodowanie

napisanie testów do kodu Hamminga

{cele, produkty, kryteria akceptacji, główne zadania itp.}

Cele: Napisanie testów sprawdzających poprawność wykonania kodu Hamminga

Produkty: Wykonujące się automatycznie przy budowaniu aplikacji testy sprawdzające, czy funkcje do zakodowania i zdekodowania informacji poprawnie działają

Kryteria akceptacji: po minimum 5 testów zakodowania i zdekodowania

napisanie testów do kodu Reeda-Solomona

{cele, produkty, kryteria akceptacji, główne zadania itp.}

Cele: Napisanie testów sprawdzających poprawność wykonania kodu Reeda-Solomona

Produkty: Wykonujące się automatycznie przy budowaniu aplikacji testy sprawdzające, czy funkcje do zakodowania i zdekodowania informacji poprawnie działają

Kryteria akceptacji: po minimum 5 testów zakodowania i zdekodowania

napisanie tłumaczenia programu na język angielski

{cele, produkty, kryteria akceptacji, główne zadania itp.}

Cele: Napisanie tłumaczenia do istniejącej polskiej wersji programu na język angielski

Produkty: Treść programu przetłumaczona na język angielski i odpowiednio wkomponowana w treść programu

Kryteria akceptacji: W całości przetłumaczony program na zrozumiały język angielski

3 Planowany podział zadań i ról w projekcie w zespole projektowym

3.1 Opis zadań planowanych do realizacji ze wskazaniem osób odpowiedzialnych

(główne zadania w poszczególnych etapach, wykazanych w harmonogramie z punktu 2)

Stworzenie repozytorium na GitHub

{wyszczególnione zadania/role, osoba odpowiedzialna/wykonawca lub wykonawcy}

Stworzenie repozytorium, nadanie wszystkim członkom praw dostępu, dodanie opisu i typowych ustawień takich jak brak możliwości pushowania do brancha main, możliwość zaakceptowania Pull Request tylko po wykonanym code review. Dzięki temu jest możliwość kontroli wersji i przywrócenia poprzedniej, jeśli zajdzie taka konieczność.

Michał Mróz

Code Review

{wyszczególnione zadania/role, osoba odpowiedzialna/wykonawca lub wykonawcy}

Wykonywanie code review do Pull Requestów na GitHubie – sprawdzenie czy kod zawiera jakieś błędy i czy wykonuje wszystkie ustalone funkcjonalności, zaakceptowanie Pull Request.

Paweł Jastrzębski, Michał Mróz, Piotr Noga

Projektowanie interfejsu użytkownika

{wyszczególnione zadania/role, osoba odpowiedzialna/wykonawca lub wykonawcy}

Stworzenie koncepcji ogólnego wyglądu aplikacji oraz implementacja tego wyglądu. Interfejs musi spełniać wymagania estetyczne.

Michał Mróz, Piotr Noga

Pisanie testów

{wyszczególnione zadania/role, osoba odpowiedzialna/wykonawca lub wykonawcy}

Napisanie testów sprawdzających poprawność napisanych funkcji, testy dotyczą zarówno procesu kodowania jak i procesu dekodowania.

Bartosz Kołakowski, Michał Mróz, Paweł Jastrzębski, Maksym Nowak

Zarządzanie zespołem

{wyszczególnione zadania/role, osoba odpowiedzialna/wykonawca lub wykonawcy}

Organizacja pracy, ustalanie terminów, dyskusja zmian w projekcie, organizowanie komunikacji grupowej, dbanie o pozytywne relacje w grupie.

Bartosz Kołakowski

Implementacja kodu nadmiarowego

{wyszczególnione zadania/role, osoba odpowiedzialna/wykonawca lub wykonawcy}

Rzeczywista implementacja kodów, zgodna z ich teoretycznymi algorytmami.

Bartosz Kołakowski, Michał Mróz, Paweł Jastrzębski, Maksym Nowak

Tworzenie wizualizacji kodu nadmiarowego

{wyszczególnione zadania/role, osoba odpowiedzialna/wykonawca lub wykonawcy}

Stworzenie wizualizacji działania kodu, pokazanie rezultatu opiekunowi projektu i wykonanie poprawek wskazanych przez opiekuna.

Bartosz Kołakowski, Michał Mróz, Paweł Jastrzębski, Maksym Nowak, Piotr Noga

Napisanie tłumaczenia na język angielski

{wyszczególnione zadania/role, osoba odpowiedzialna/wykonawca lub wykonawcy}

Przetłumaczenie programu tak, aby był zrozumiały dla osób, które posługują się językiem angielskim. Zakłada się, że funkcjonalność programu będzie tak samo dostępna zarówno w języku polskim jak i angielskim.

Piotr Noga

Pisanie dokumentów

{wyszczególnione zadania/role, osoba odpowiedzialna/wykonawca lub wykonawcy}

Napisanie wszystkich dokumentów wymaganych do zaliczenia projektu grupowego; harmonogramu, dokumentacji, raportu, plakatu, informacji o projekcie

Bartosz Kołakowski, Paweł Jastrzębski

Stworzenie prezentacji

{wyszczególnione zadania/role, osoba odpowiedzialna/wykonawca lub wykonawcy}

Stworzenie prezentacji na temat postępu projektu, która będzie zaprezentowana na seminarium katedralnym

Bartosz Kołakowski, Paweł Jastrzębski, Maksym Nowak

4 Wymagania dla produktu i kryteria akceptacji

4.1 Ogólny opis planowanego produktu

{sporządzić ogólną charakterystykę produktu, co to ma być, do czego służy, jakie główne funkcje ma realizować itp.; w semestrze 2 należy zaktualizować informacje, jeśli wystąpiły zmiany}

Celem projektu jest napisanie narzędzia wspomagającego proces dydaktyczny. Narzędziem ma być aplikacja wizualizująca zagadnienia związane z kodowaniem nadmiarowym. Aplikacja ma obrazowo prezentować klasyfikację kodów nadmiarowych, proces kodowania i dekodowania wybranych kodów nadmiarowych, zysk kodowania, różne zagadnienia związane z tematem, oraz pozwalać na przeprowadzanie eksperymentów. Aplikacja ma być dostępna w wielu językach.

4.2 Wymagania minimalne dla produktu

{opisać jakie są minimalne wymagania jakościowe dla produktu, spróbować określić metody zbadania wymagań minimalnych, podać metody weryfikacji; w semestrze II należy zweryfikować i zaktualizować informacje z semestru I, jeśli wystąpiły zmiany}

Trzeba wykonać wizualizację przynajmniej dwóch kodów nadmiarowych – kodu Hamminga i kodu Reeda-Solomona. Aplikacja musi mieć estetyczny graficzny interfejs użytkownika i musi być uruchamiana w systemie MS Windows. Aplikacja musi być przetłumaczona na język angielski. Aplikacja musi zapewniać możliwość podania danych przez użytkownika.

4.3 Warunki odbioru

{z punktu widzenia umowy - kiedy uznamy, że projekt zakończył się sukcesem – testy kwalifikacyjne, spełnienie wymagań, (warunki techniczne, prawne, finansowe; w semestrze II należy zweryfikować i zaktualizować informacje z semestru I, jeśli wystąpiły zmiany)}

Spełnienie wymagań projektu założonych przez opiekuna do końca czasu przeznaczonego na wykonanie, czyli końca drugiego semestru.

Akceptacja projektu przez opiekuna projektu.

5 Postanowienia

5.1 Postanowienia w zakresie zmian w stosunku do pierwotnego planu i zakresu prac

{jeżeli występują jakieś zmiany w stosunku do planu pierwotnego, to należy je wskazać, np. jeżeli harmonogram był utworzony w całości na I semestrze, ale od razu na 2 semestry i następują jakieś zmiany, to należy je wskazać, jeżeli nie ma postanowień to należy wpisać „nie dotyczy”}

Na początku kodowanie Reeda-Solomona (oraz wizualizacja i napisanie do nich testów) miały być zrealizowane w pierwszym semestrze, jednak ze względu na skupienie się na jakości wykonania kodowanie Reeda-Solomona zostało przełożone na drugi semestr.

5.2 Inne postanowienia

{jeżeli są}