



**POLITECHNIKA
GDAŃSKA**

WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI I INFORMATYKI



Imię i nazwisko studenta: Adam Borowski
Nr albumu: 137252
Studia drugiego stopnia
Forma studiów: stacjonarne
Kierunek studiów: Informatyka
Specjalność/profil: Aplikacje rozproszone i
systemy internetowe

PRACA DYPLOMOWA MAGISTERSKA

Tytuł pracy w języku polskim: Stworzenie i rozpowszechnienie otwartego webowego narzędzia do współpracy przy edycji danych pomiarowych w czasie rzeczywistym

Tytuł pracy w języku angielskim: Creation and popularization of web-based open source tool for effective real-time collaboration on measurement data.

Potwierdzenie przyjęcia pracy	
Opiekun pracy	Kierownik Katedry/Zakładu
podpis	podpis
dr hab. inż. Jerzy Balicki, prof. nadzw. PG	

Data oddania pracy do dziekanatu:



**POLITECHNIKA
GDAŃSKA**

WYDZIAŁ ELEKTRONIKI,
TELEKOMUNIKACJI I INFORMATYKI



OŚWIADCZENIE

Imię i nazwisko: Adam Borowski
Data i miejsce urodzenia: 24.05.1991, Gdańsk
Nr albumu: 137252
Wydział: Wydział Elektroniki, Telekomunikacji i Informatyki
Kierunek: informatyka
Poziom studiów: II stopnia
Forma studiów: stacjonarne

Ja, niżej podpisany(a), wyrażam zgodę/nie wyrażam zgody* na korzystanie z mojej pracy dyplomowej zatytułowanej: Stworzenie i rozpowszechnienie otwartego webowego narzędzia do współpracy przy edycji danych pomiarowych w czasie rzeczywistym do celów naukowych lub dydaktycznych.¹

Gdańsk, dnia

.....
podpis studenta

Świadomy(a) odpowiedzialności karnej z tytułu naruszenia przepisów ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz. U. z 2006 r., nr 90, poz. 631) i konsekwencji dyscyplinarnych określonych w ustawie Prawo o szkolnictwie wyższym (Dz. U. z 2012 r., poz. 572 z późn. zm.),² a także odpowiedzialności cywilno-prawnej oświadczam, że przedkładana praca dyplomowa została opracowana przeze mnie samodzielnie.

Niniejsza(y) praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadaniem tytułu zawodowego.

Wszystkie informacje umieszczone w ww. pracy dyplomowej, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami zgodnie z art. 34 ustawy o prawie autorskim i prawach pokrewnych.

Potwierdzam zgodność niniejszej wersji pracy dyplomowej z załączoną wersją elektroniczną.

Gdańsk, dnia

.....
podpis studenta

Upoważniam Politechnikę Gdańską do umieszczenia ww. pracy dyplomowej w wersji elektronicznej w otwartym, cyfrowym repozytorium instytucjonalnym Politechniki Gdańskiej oraz poddawania jej procesom weryfikacji i ochrony przed przywłaszczeniem jej autorstwa.

Gdańsk, dnia

.....
podpis studenta

*) niepotrzebne skreślić

¹ Zarządzenie Rektora Politechniki Gdańskiej nr 34/2009 z 9 listopada 2009 r., załącznik nr 8 do instrukcji archiwalnej PG.

² Ustawa z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym:

Art. 214 ustęp 4. W razie podejrzenia popełnienia przez studenta czynu podlegającego na przypisaniu sobie autorstwa istotnego fragmentu lub innych elementów cudzego utworu rektor niezwłocznie poleca przeprowadzenie postępowania wyjaśniającego.

Art. 214 ustęp 6. Jeżeli w wyniku postępowania wyjaśniającego zebrany materiał potwierdza popełnienie czynu, o którym mowa w ust. 4, rektor wstrzymuje postępowanie o nadanie tytułu zawodowego do czasu wydania orzeczenia przez komisję dyscyplinarną oraz składa zawiadomienie o popełnieniu przestępstwa.

ABSTRACT

The **Abstract** should include a definition of the scientific or practical problem to be solved, the aim and scope of the thesis, the applied methods of research, the results and the most important conclusions.

If the thesis is written by more than one author, the **Abstract** must specify what each author has prepared and contributed. Moreover, the particular authorship of each chapter and subchapter must also be specified (see **Table of Contents**). If a writer is named as the author of a particular subchapter it should be assumed that all the items in the subchapter are the work of that particular writer. For example, if the thesis is co-authored by writer A and writer B, in the Table of Contents it may be stated that writer A co-authored chapters 1 and 7, individually authored chapter 2 as well as subchapters 3.1 and 4.2, etc. A writer B, on turn, may be named as the co-author of chapters 1 and 7 as well as the individual author of subchapters 3.2 and 4.1, etc.

Keywords:

Field of science and technology in accordance with OECD requirements: <field>, <technology>,...

STRESZCZENIE

TOOO Analiza istniejących projektów na githubie. sukcesy i porażki Ile projektów jest **** a ile upadło. Jak wygląda historia commitów jednych i drugich.

Teza1: najpierw jest super pomysł, nie ważne jak wygląda kod, dokumentacja, strona domowa, dopiero potem, jak ludzie chcą kontrybuować, to jest tzw wersja 2.0, gdzie jest dużo tooli i wszystko smiga

Teza2: nawet pomysł może nie być super hiper, ale tool jest wiarygodny spełnia standardy, jest dostępny z bowera itd. więc użytkownicy chcą to lubia.

The **Streszczenie** should correspond to the **Abstract** and includes the same key elements. It should be prepared according to faculty rules. The **Streszczenie** is a chapter written only by Polish authors who decide to write the diploma thesis in English and it is not applicable to foreign students.

Słowa kluczowe: [keywords] Zażółć gęślą jaźń.

Dziedzina nauki i techniki, zgodnie z wymogami OECD: <dziedzina>, <technika>, ... [field of science and technology in accordance with OECD requirements: <field>, <technology>, ...]

SPIS TREŚCI

List of important symbols and abbreviations	6
1. Wstęp	7
1.1. Cel pracy	9
1.1.1. Cel biznesowy i Open Source	9
1.1.2. Produkt pracy	9
1.1.3. Charakterystyka Open Source	9
1.2. Motywacja	9
1.2.1. Monitoring środowiska	9
1.2.2. Stan wyjściowy	9
1.3. Główne wyzwania	9
1.4. Aspekt badawczy	9
1.5. Wykorzystane narzędzia	9
1.6. Wykorzystane wzorce projektowe	9
2. Praca w ujęciu biznesowym	10
2.1. Definicja klienta	10
2.2. Analiza istniejących rozwiązań	10
2.3. Wymagania biznesowe	10
2.3.1. Wymagania funkcjonalne	10
2.3.2. Wymagania pozafunkcjonalne	10
3. Praca w ujęciu programistycznym Open Source	11
3.1. Definicja klienta	11
3.2. Analiza istniejących rozwiązań	11
3.3. Wymagania programistyczne w kontekście Open Source	11
3.4. Repozytorium GitHub	11
3.5. Architektura Przyjazna deweloperom	11
4. Opis architektury prezentowanego rozwiązania	12
5. Title of chapter 2	13
5.1. Subchapter title	13
5.1.1. Subchapter section heading	14
5.2. Subchapter heading	14
6. Summary	15
Bibliography	16
List of Figures	17
List of Tables	18
Dodatek A. Title of Appendix A	19
Dodatek B.	20

OZNACZENIA, SYMBOLE I SKRÓTY

e	—	uncertainty of measurement
f	—	frequency [Hz]
i, j, l, m	—	indexes
k	—	Boltzmann constant $1,38 \cdot 10^{-23}$ Ws/K
T	—	time of measurement [s]
CDM	—	Context Driven Model
SOA	—	Service Oriented Architecture

1WSTĘP

Produktem pracy będzie biblioteka w języku JavaScript do wykorzystania w zaawansowanych aplikacjach webowych w przeglądarce internetowej.

Kilka uwag do tematu pracy magisterskiej:

- Pomysł na temat pracy jest związany z moim doświadczeniem zawodowym zdobytym w firmie DAC System, a problemy technologiczne wskazane poniżej istotnie wpływają na efektywność pracy monitoringu jakości powietrza w Wojewódzkich Inspektoratów Ochrony Środowiska, czyli użytkowników produktu, który rozwijałem we tej firmie.
- Część poruszonych problemów została przeze mnie rozwiązana w ramach pracy w firmie DAC System, i zgodnie z sugestią firmy włączę je do swojej pracy.
- Mowa tutaj o danych pomiarowych, czyli danych zbieranych przez urządzenia pomiarowe oznaczone stemplem czasowym, zbieranym przez wiele lat co kilka sekund. Jest to zagadnienie związane z tzw. Big Data
- Produktem końcowym jest arkusz danych pomiarowych edytowany analogicznie do edycji w programie Excel (zdjęcie poniżej) i wykres liniowy pomagający wyłapać błędne pomiary. Oba produkty dostosowane będą do płynnej obsługi wielkich zbiorów danych pomiarowych.
- Główne wyzwania do podjęcia:
 - Zaprojektowanie protokołu komunikacyjnego klient-serwer opartego na technologii AJAX, którego zadaniem jest ukrycie przed użytkownikiem faktu, że pracuje on tylko na pewnym wycinku danych. Z tego protokołu będą korzystały komponenty graficznego interfejsu użytkownika.
 - Optymalizacja procesu renderowania komponentu arkusza danych, by wydajnie obsługiwać lokalne zbiory danych niezależnie od ich rozmiaru.
 - Zaprojektowanie komponentu wykresów liniowych, pozwalającego na płynne przeglądanie danych pomiarowych, minimalizującego czas oczekiwania użytkownika na dane.
- Przypadki użycia które postawiły te wyzwania technologiczne:
 - Użytkownik chce płynnie przeglądać dane z dowolnego zakresu czasowego
 - Kiedy użytkownik powraca do danych już odwiedzonych (np. przewijając wykres w przód lub w tył), dane powinny być odtworzone z pamięci lokalnej.
 - Niemożliwe jest pobranie wszystkich danych do klienta na jego żądanie
 - Klient powinien dokonywać predykcji, które fragmenty danych będą za chwilę potrzebne, aby minimalizować czas oczekiwania na pobranie następnego kawałka danych
 - Klient powinien dbać o zwalnianie zasobów z pamięci na podstawie informacji, które fragmenty danych były użyte najdawniej / najrzadziej.
- Rozwiązanie protokołu komunikacji podobne jest do tego stosowanego w mechanizmie używanym przez Mapy Google, w kontekście ładowania kafelków mapy.

Oprócz wytworzenia narzędzia w JavaScript, chcę skupić się na tym, jak przygotować swoją pracę dla innych, by z niej mogli skorzystać inni użytkownicy serwisu <http://github.com/>. Chodzi o czytelność i wiarygodność projektu. Mam tutaj na myśli:

- wyodrębnienie mniejszych, samodzielnych i użytecznych modułów, które rozwiązują pojedyncze problemy programistyczne
- wersjonowanie zgodnie z konwencją Semantic Versioning 2.0.0 (<http://semver.org/>)
- jasno zdefiniowane i opisane interfejsy programistyczne

- korzystanie ze standardów - dostępność modułów w czasie wykonania: AMD (np. <http://requirejs.org/>) oraz w środowisku wytwórczym: <http://bower.io/> lub <http://npmjs.com/>
- dbałość o kod (m.in. brak polskich wstawek w komentarzach), staranne oznaczanie zmian w repozytorium GIT
- jakość: pokrycie testami jednostkowymi i funkcjonalnymi, ciągła integracja (ang. continuous integration)
- dostosowanie się do obecnych trendów programowania w JavaScript: EcmaScript6, BabelJS, Gulp

»»»

Niniejsza praca będzie poświęcona głównie ideologii Open Source, czyli tej, w której twórcy oprogramowania dzielą się swoim dziełem udostępniając wszystkie zasoby źródłowe, przyczyniając się tym samym do rozpowszechniania wiedzy i praktyk.

Osoby zainteresowane tym dobrem wspólnym skupiają się wokół specjalistycznych portali internetowych tworząc tym samym specyficzną społeczność. Jedną z nich jest społeczność portalu **github.com**, który jest obecnie powszechnie znanym zbiorem repozytoriów, miejscem, w którym utrzymywane są najważniejsze i najpopularniejsze biblioteki programistyczne.

Korzyści płynące z korzystania z darmowych, utrzymywanych i rozwijanych, wiarygodnych projektów są zgoła oczywiste. Inaczej jest z korzyściami płynącymi z twórczego wkładu w Otwarte Oprogramowanie. Jedni chcą móc się pochwalić autorstwem powszechnego narzędzia, inni chcą mieć wartościowy wpis w życiorysie, robią to w ramach pracy, a jeszcze inni czują po prostu potrzebę podzielenia się swoim dziełem z innymi. «< aspekt zopensourcownani W tej pracy postawię się w roli programisty chcącego utworzyć nowy projekt open source, który będzie współrozwijany przez społeczność GitHuba, zyska na popularności i nabierze własnego rytmu. Niestety, można się domyśleć, że do osiągnięcia opensourcowego sukcesu nie wystarczy załadować swój "idealny" projekt do repozytorium GitHuba. Nawet najlepsze pomysły, którymi chce się podzielić twórczy programista, są skazane na brak zainteresowania, jeśli nie są dostosowane do wymagań innych użytkowników. Aby sprawdzić, czym powinien charakteryzować się nowy, nieznany jeszcze projekt, będę musiał postawić się również w roli programisty, który przypadkiem znalazł ciekawą bibliotekę nadającą się do jego projektu, ale która w ogóle nie jest znana, co wiąże się zwiększonym ryzykiem porażki całego projektu. Tylko wtedy, gdy sam użyje tej biblioteki, może stwierdzić, że warto poświęcić swój czas na jej rozwijanie.

Postaram się znaleźć dobre odpowiedzi na poniższe pytania:

- kiedy bibliotekę otwartoźródłową społeczność uważa za godną użycia,
- jakie są zagrożenia płynące z użycia niewiarygodnej biblioteki,
- czego wymaga się od bibliotek rozwijanych przez innych.

Zanim rozpocznę tworzenie projektu open source na portalu github.com, przeprowadzę krótką analizę, by dowiedzieć się, czym charakteryzowały się w swojej początkowej fazie projekty na tym portalu, które obecnie zyskały na popularności. Ponieważ biblioteka, którą chcę tworzyć, będzie napisana w języku JavaScript, w analizie ograniczę się do małych bibliotek również napisanych w tym języku. Dzięki temu, z zebranych spostrzeżeń będę mógł wysnuć wnioski pomocne przy planowaniu projektu.

Na wstępie warto zaznaczyć, że pomysł realizowany w projekcie powinien być innowacyjny, nie powinien być tylko alternatywą istniejącego rozwiązania, gdyż można się domyśleć, że ludzie mając porównanie z popularną biblioteką, będą kierowali się raczej tym ostatnim w kryteriach swojego wyboru. »»»odtąd prawdziwy wstęp Projekt będzie miał na celu wytworzenie narzędzia programistycznego rozwiązującego problemy występujące w bliskiej mi branży telemetrii i monitoringu środowiska. Problemy te poznałem zdobywając doświadczenie w firmie DAC System,

gdzie brałem udział w wytwarzaniu systemu CAS14, tzw. **grubego klienta** systemu monitoringu jakości powietrza dla ośmiu Wojewódzkich Inspektoratów Ochrony Środowiska, zamawianych przez Główny Inspektorat Ochrony Środowiska w zamówieniu *ZP/DM/0811-07-EOG/01/2013/MZ*.

Część tych problemów rozwiązałem w tamtym systemie, dlatego za zgodą i jednocześnie sugestią ze strony firmy DAC System, przeniosę część rozwiązań do narzędzia otwartego.

1.1. Cel pracy

1.1.1. Cel biznesowy i Open Source

Cel biznesowy

Cel Open Source

1.1.2. Produkt pracy

1.1.3. Charakterystyka Open Source

1.2. Motywacja

1.2.1. Monitoring środowiska

1.2.2. Stan wyjściowy

1.3. Główne wyzwania

1.4. Aspekt badawczy

1.5. Wykorzystane narzędzia

1.6. Wykorzystane wzorce projektowe

2PRACA W UJĘCIU BIZNESOWYM

2.1. Definicja klienta

Charakterystyka pracy w WIOŚ
Co robią, co walidują, jakieś normy

2.2. Analiza istniejących rozwiązań

2.3. Wymagania biznesowe

2.3.1. Wymagania funkcjonalne

2.3.2. Wymagania pozafunkcjonalne

3PRACA W UJĘCIU PROGRAMISTYCZNYM OPEN SOURCE

3.1. Definicja klienta

3.2. Analiza istniejących rozwiązań

3.3. Wymagania programistyczne w kontekście Open Source

3.4. Repozytorium GitHub

3.5. Architektura Przyjazna deweloperom

4OPIS ARCHITEKTURY PREZENTOWANEGO ROZWIĄZANIA

5TITLE OF CHAPTER 2

The title of a subsequent chapter should always appear on a new page. Every paragraph containing the title of a chapter or subchapter has the following intervals:

- top 12 pt.,
- bottom 6 pt.

Here is some text to get rid of Underfull Badbox. Here is some text to get rid of Underfull Badbox. Here is some text to get rid of Underfull Badbox. Here is some text to get rid of Underfull Badbox. Here is some text to get rid of Underfull Badbox. Here is some text to get rid of Underfull Badbox. Here is some text to get rid of Underfull Badbox. Here is some text to get rid of Underfull Badbox. Here is some text to get rid of Underfull Badbox. Here is some text to get rid of Underfull Badbox. Here is some text to get rid of Underfull Badbox. Here is some text to get rid of Underfull Badbox.

5.1. Subchapter title

References in the text to literature should show in square brackets their reference number from the **Bibliography** [1] or the name of the author/s and year of publication [Kowalski J., 2002] and page number in the case of citations [Kowalski J., 2002, p. 3].

The **Bibliography** should be arranged alphabetically or in the order items are mentioned in the chapter. If several items of literature are referred to at one point in the text, a dash may be used in the square brackets [1-5] if they are in consecutive order in the **Bibliography**, or if their order is not consecutive, using commas, as follows: [1, 3, 5].

An example of bibliographic entries is shown in the Bibliography chapter.

Figures are numbered in the order in which they appear in the chapter. In the caption below a figure, the number of the figure is preceded by the abbreviation 'Fig.' and the appropriate chapter number, e.g. 'Fig. 2.1.'

Figures appearing in the thesis should be centred. The spacing in a paragraph containing a figure should be:

- top 12 pt,
- bottom 0 pt.

Every figure must be referred to in the text, for example as follows: 'Fig. 5.1. presents. . . '.

If the thesis is written in English, all the tables and figures must also be presented in the English language.

An example of a properly presented figure and caption is provided below.



Rys. 5.1. Emblem of Gdańsk University of Technology.

The caption below the figure should be centered, have a 9 pt. font size and end with a full stop. The top margin should be 6 pt., the bottom margin 12 pt. and the spacing should be single.

The first paragraph below a figure should have a top margin of 12 pt.

5.1.1. Subchapter section heading

Every paragraph containing a subchapter section heading should have the following spacing:

- top 12 pt.,
- bottom 6 pt.

5.2. Subchapter heading

Variables should be written in italics, e.g. x , ni , $ni + 1$, whereas symbols denoting vectors or matrices should be written in bold, e.g. \mathbf{v} , \mathbf{A} . The minus sign should directly precede the number without a space, e.g. -20 . There must always be a space between a numerical value and unit of measurement, e.g. 1 V or 10 km.

Preferably, the insertion of equations (INSERT -> Equation) directly into the text should be avoided. If there is a possibility of legibly inserting an equation in one line, it should be done as in the following examples: $1/2t^2$ or e^{2x+1} . This should be immediately followed by an explanation as to what the particular symbols denote, e.g. where: t — time [s].

Equations that might become illegible when written in a single line should be written in a separate paragraph, e.g.

$$s = v_0 \cdot t + \frac{a \cdot t^2}{2} \quad (5.1)$$

where:

- | | | |
|-------|---|---|
| s | — | linear displacement with constant acceleration [m], |
| v_0 | — | initial velocity [m/s], |
| t | — | time of object motion [s], |
| a | — | acceleration [m/s ²]. |

Every equation should be centered. Its number should be preceded by the chapter number and a dot, in parentheses. Every equation has to be referred to in the text, e.g. 'Equation (5.1) allows us to estimate. . . '.

Any footnotes should appear below a line at the bottom of the page,¹ and there numbering should be consecutive throughout the thesis. To make footnotes use NUMBERING OPTIONS -> add footnote.

¹Footnotes should be written using Arial 9 pt.

6SUMMARY

Every diploma thesis must include a chapter entitled **Summary**. It should appear before the **Bibliography** and include a review of the main points of the thesis and/or obtained results. The chapter should also state what should be realized if research into the subject of the thesis is continued.

Bibliografia

- [1] J.L. Hennessy, D.A. Patterson, and K. Asanović. *Computer Architecture: A Quantitative Approach*. Computer Architecture: A Quantitative Approach. Morgan Kaufmann/Elsevier, 2012.

SPIS RYSUNKÓW

5.1. Emblem of Gdańsk University of Technology.	13
--	----

SPIS TABLIC

DODATEKATITLE OF APPENDIX A

Appendices should be consecutively denoted with letters of the alphabet. An Appendix should include necessary supplementary data, e.g. calculations or schematic diagrams.

DODATEKBTITLE OF APPENDIX B

Detailed guidelines regarding the content of Appendices should be established by the faculty, taking into account the specificity of a given field and course of study.