

Analysis of the application for the DFD authoring usage possibilities

Analiza możliwości wykorzystania aplikacji do tworzenia diagramów DFD

Marek Pieczykolan*, Marcin Badurowicz

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20-618 Lublin, Poland

Abstract

This article analyzes the possibility of using application to create data flow diagrams, by using QSEE Superlite and the application created in Flutter framework. The purpose of the article is to make comparative analysis of the time needed to create a data flow diagram, as well as to evaluate users in terms of interface transparency and freedom of action in each application. Two research hypotheses have been formulated: “DFD Maker enables faster creation of data flow diagrams compared to QSEE Superlite” and “DFD Maker has a more user-friendly interface compared to QSEE Superlite, allowing for easier creation of data flow diagrams”, which were confirmed after analyzing the results of the conducted study.

Keywords: data flow diagrams; DFD; Flutter

Streszczenie

Artykuł dotyczy analizy możliwości wykorzystania aplikacji do tworzenia diagramów przepływu danych przy wykorzystaniu QSEE Superlite oraz aplikacji stworzonej we frameworku Flutter. Celem artykułu jest dokonanie analizy porównawczej czasu potrzebnego do stworzenia diagramu przepływu danych, a także oceny użytkowników pod kątem przejrzystości interfejsu oraz swobody działania w każdej z aplikacji. Postawiono dwie hipotezy badawcze: „DFD Maker pozwala na szybsze tworzenie diagramów przepływu danych w porównaniu do QSEE Superlite” oraz „DFD Maker ma przyjaźniejszy interfejs dla użytkownika w porównaniu do QSEE Superlite co pozwala na łatwiejsze tworzenie diagramów przepływu danych”, które po analizie wyników, przeprowadzonych badań, zostały udowodnione.

Słowa kluczowe: diagramy przepływu danych; DFD; Flutter

*Corresponding author

Email address: marek.pieczykolan@pollub.edu.pl (M. Pieczykolan)

Published under Creative Common License (CC BY-SA v4.0)

1. Wstęp

Ważnym aspektem przed rozpoczęciem procesu tworzenia oprogramowania jest odpowiednie zaprojektowanie systemu informatycznego. Wcześniejsza analiza wymagań, określenie reakcji systemu na różne zdarzenia pozwala na skrócenie późniejszego etapu, w którym system powstaje. Co przy obecnie dużej konkurencyjności rynku technologicznego, może okazać się kluczowe. Kosztowność rozwoju bardzo dużych projektów dodatkowo podkreśla jak ważnym aspektem jest metodyczne oraz stopniowe podejście do tego zagadnienia. Jednym z ważnych elementów przy procesie projektowaniu systemu informatycznego, szczególnie w biznesie, jest stworzenie diagramów przepływu danych, które pomogą w określeniu specyfikacji dla oprogramowania, a także ułatwią zdefiniowanie wymagań systemowych oraz rozpoznawanie i rozwiązywanie błędów [1].

Diagramy przepływu danych (ang. Data Flow Diagram - DFD) dzięki wizualnemu przedstawieniu modeli logicznych, a także transformacji oraz przepływu danych, służą jako ustrukturyzowana metoda do analizy i projektowania. Poprzez modelowanie przepływu danych, diagramy, wspierają dekompozycję, która pomaga w zilustrowaniu funkcji oraz przepływów

między nimi. DFD zawierają charakterystyki takie jak, wsparcie etapu wymagań projektu oraz jego analizę, dopuszczenie procedur asynchronicznych oraz równoległych, tworzenie diagramów z adnotacjami. Ze względu na swoje przeznaczenie diagramy DFD nie mogą przedstawić kolejności wykonywania działań, dlatego nie są metodami, których można użyć do modelowania procesów jak i procedur [2].

Proces tworzenia diagramów przepływu danych wymaga przestrzegania pewnych zasad, które pozwolą zminimalizować liczbę błędów, a także pozwalają na poprawne sprawdzenie spójności pomiędzy zależnymi od siebie diagramami. Jest to ważne w kontekście czasochłonności całego procesu. Diagramy DFD można podzielić na trzy główne stopnie szczegółowości: diagram kontekstowy, diagram systemowy oraz diagram szczegółowy. Diagram kontekstowy który jest diagramem najwyższego poziomu, zawiera granice systemu informatycznego oraz pozwala na określenie obrotów oraz źródła danych. Następnym poziomem jest diagram systemowy zwany diagramem poziomu zerowego, zawiera on główne funkcje oraz magazyny danych, które są powiązane z funkcjami, zaprojektowanego systemu. Ostatnim stopniem jest diagram szczegółowy n elementowy, jego zadaniem jest przedstawienie przepływu danych w procesach. Podczas tworzenia DFD powinno się dekomponować system do

etapu, w którym będzie możliwe dostarczenie szczegółowego opisu procesu [3].

Celem niniejszego artykułu jest porównanie dwóch aplikacji do tworzenia diagramów przepływu danych. Pierwszą z nich jest QSEE Superlite, która umożliwia ona na tworzenie wielu rodzajów diagramów w tym DFD. Wszystkie badania zostaną przeprowadzone w stylu Metodologii analizy i projektowania systemów strukturalnych (ang. Structured Systems Analysis and Design Methodology - SSADM). Drugą jest aplikacja stworzona przy pomocy platformy Flutter nazwana DFD Maker. Kształt komponentów jest niemal identyczny jak w przypadku QSEE. W celu potwierdzenia hipotez, uczestnicy procesu analizy stworzą diagramy przepływu danych w dwóch aplikacjach, zostanie zmierzony czas potrzebny do stworzenia każdego z nich oraz uczestnicy wypełnią ankietę. Na tej podstawie zostały postawione niniejsze hipotezy:

- H1. DFD Maker pozwala na szybsze tworzenie diagramów przepływu danych w porównaniu do QSEE Superlite.
- H2. DFD Maker ma przyjaźniejszy interfejs dla użytkownika w porównaniu do QSEE Superlite co pozwala na łatwiejsze tworzenie diagramów przepływu danych.

2. Przegląd literatury

Środowisko naukowe dostarcza wiele artykułów opisujących zastosowania diagramów przepływu danych. Omawiają one nie tylko stronę teoretyczną, ale odnoszą się także do konkretnych przykładów użycia DFD przy wstępnym procesie planowania oprogramowania, prezentując wszelkie zalety jak i wady diagramów.

W pracy [4] zbadano zastosowanie diagramów przepływu danych w analizie systemowej dla procesu realizacji zamówień. Skupiono się w nim na stworzeniu modelu, który zobrazował występowanie kluczowych miejsc powodujących opóźnienia, a także ukazał najważniejsze etapy.

Autorzy w pracy [5] stworzyli model obrazujący zachodzące procesy w przedsiębiorstwie w postaci diagramów przepływu danych. Wykazano w niej jak wszechstronnym narzędziem są diagramy przepływu danych. Ich odpowiednie zastosowanie pozwala na zobrazowanie przepływu danych w taki sposób, aby określały one najsłabsze punkty w przepływie, pozwalając na wdrożenie zmian, które zniwelują owe punkty.

W pracy [6], w szczegółowy sposób opisano diagramy przepływu danych. Każdy z elementów został przedstawiony, a także dokładnie wytłumaczono zadania, które spełnia każdy z nich. Autor zaznaczył różnice pomiędzy poszczególnymi poziomami DFD, wyszczególniając podział na trzy kategorie: diagram kontekstowy; diagram systemowy; diagram szczegółowy, który zawiera niższe poziomy w modelu.

Każda z kategorii została poza szczegółowym wytłumaczeniem zawiera rady, którymi należy się kierować podczas tworzenia diagramów przepływu danych.

W pracy [7] stworzono środowisko, które na podstawie diagramów przepływu danych w połączeniu z adnotacjami pozwala na przeprowadzenie symulacji. Symulacja obrazuje wydajność zaprojektowanego systemu, dając możliwość pełniejszego wykorzystania DFD. Wyniki wskazują jak przydatnym narzędziem mogą okazać się diagramy w planowaniu projektu przy odpowiednim ich wykorzystaniu.

W artykule [8] dostarczono rygorystycznych ram formalnych, które wskazują w jaki sposób mają być opisywane diagramy poprzez etykiety celu, a także podpisy prywatności. Jest to rozwinięcie możliwości jakie prezentują diagramy przepływu danych, ukazując jak wszechstronnym narzędziem mogą się one okazać i na jak wielu płaszczyznach mogą pomóc w projektowaniu oraz tworzeniu oprogramowania.

W pracy [9] ukazano sposób dołączania semantyki do diagramów przepływu danych. Przedstawiono sposób przekształcania DFD na sieci Petriego. Sposób ten przedstawia aspekty synchronizacji diagramów przepływu danych.

W pracy [10] przedstawiono zalety jakie niesie wykorzystanie diagramów przepływu danych w kontekście wygody użytkownika. Odpowiednie użycie DFD może zwiększyć komfort z korzystania z aplikacji, poprzez wcześniejsze dostarczenie informacji o przepływie danych pomiędzy poszczególnymi elementami oprogramowania. Dzięki wiedzy na temat przepływu informacji w aplikacji, można poprawić funkcjonalność, dostosowując elementy, w sposób najbardziej przyjazny użytkownikowi.

Autorzy w pracy [11], używają diagramów przepływu danych do ukazania wzorców użytkownika oraz celów bankowości internetowej. Zastosowanie DFD w artykule obrazuje jak wiele zalet niesie ze sobą użycie tego narzędzia. Najważniejszą zaletą jest ilość dostarczanych informacji w formie wizualnej, która ułatwia zrozumienie przepływu danych, a także w jaki sposób wykonywane są kolejne procesy, które tworzą całość oprogramowania bankowego. Diagramy przepływu danych dostarczyć mogą również informacji na temat potencjalnych zagrożeń oraz użyteczności w wizualizowanym modelu. Wczesne wykrycie zagrożeń może pozwolić na zmniejszenie ilości problemów, które napotkają programiści podczas implementacji funkcjonalności systemu informatycznego.

Przegląd literatury zaprezentował jak ważnym oraz przydatnym narzędziem są diagramy przepływu danych. Odpowiednie stworzenie modelu oprogramowania przy pomocy DFD może nie tylko pozwolić na lepsze zrozumienie funkcjonowania systemu informatycznego, ale również wskazać miejsca, które są szczególnie narażone na występowanie błędów. Dlatego ważnym jest użycie odpowiedniego narzędzia, które pozwoli na szybkie oraz intuicyjne tworzenie diagramów. Jest to szczególnie istotne w przypadku dużych

i skomplikowanych systemów, gdzie mogą wystąpić znaczące problemy wymagające wprowadzenia zmian w modelu.

3. Aplikacje testowe

W analizie danych oraz projektowaniu systemów informatycznych, kluczową rolę odgrywa wybór odpowiedniego narzędzia, które wspomogę proces tworzenia diagramów przepływu danych. Wraz z postępem technologii, który objawia się poprzez rosnącą wydajność sprzętu, wzrasta też poziom złożoności oprogramowania. Powoduje to potrzebę tworzenia bardziej skomplikowanych DFD. Przy dużym stopniu złożoności, ważne jest, aby narzędzie wspierało użytkownika w sposób pozwalający na jak największą optymalizację pracy, jak i ułatwienie jej. W badaniu zostały użyte dwie aplikacje, które pozwalają na tworzenie diagramów przepływu danych. Porównano ich możliwości, a także przyjazność dla użytkowników.

3.1. QSEE Superlite

Pierwszą aplikacją, której możliwości zostały sprawdzone jest QSEE Superlite w wersji 1.1.2. Aplikacja pozwala na tworzenie diagramów DFD w dwóch różnych stylach.

Pierwszym stylem jest Structured Systems Analysis and Design Method, diagramy w tym stylu obrazują przepływ danych pomiędzy procesami, zbiornikami danych, a także obiektami zewnętrznymi. Narzędzie multi-CASE wymusza stosowanie szeregu reguł, które blokują możliwość wystąpienia niespójności w modelu. Dzięki zapewnionej abstrakcji, umożliwiającej podzielenie systemu na zarządzalne segmenty, poruszanie się, a także analiza modelu jest prosta dla użytkownika.

Drugim stylem jest Real Time – Yourdon, który pozwala na modelowanie diagramów przepływu danych, ale dodatkowo wprowadza obsługę systemów czasu rzeczywistego. Styl także, jak poprzedni, bazuje na narzędziu multi-CASE, przez co również posiada reguły blokujące możliwość wystąpienia niespójności w modelu oraz wcześniej wymienione aspekty w stylu Structured Systems Analysis and Design Method.

W badaniu do analizy został wybrany styl Structured Systems Analysis and Design Method ze względu na swoją prostotę, a także większe podobieństwo diagramów do względem tych tworzonych w następnej aplikacji.

Aplikacja QSEE Superlite została wybrana ze względu na swoją popularność. Obecnie nadal jest używana do tworzenia diagramów DFD.

3.2. Aplikacja Flutterowa (DFD Maker)

Flutter jest to framework otwartoźródłowy, stworzony oraz rozwijany przez firmę Google. Jako narzędzie wieloplatformowe pozwala na tworzenie aplikacji na wiele platform przy pomocy jednego kodu źródłowego. Obiektowy język programowania Dart, na którym jest oparty Flutter kompilowany jest do kodu natywnego dla każdej z platform, co przekłada się na zbliżoną

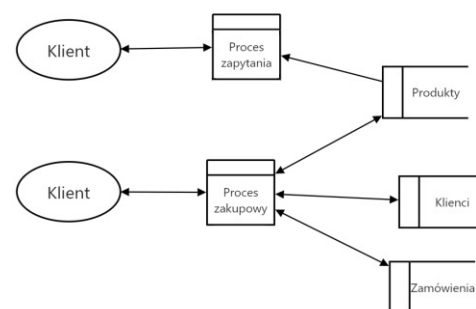
wydajność do aplikacji stworzonych w językach natywnych [12]. Bogata biblioteka, tworzona nie tylko przez twórców platformy, pozwala na szybsze tworzenie aplikacji, a także ułatwia cały proces. Ze względu na swoje zalety ten właśnie framework posłużył jako narzędzie do stworzenia aplikacji do tworzenia diagramów DFD.

Stworzoną aplikację nazwano DFD Maker. Głównym założeniem podczas tworzenia aplikacji było zaadaptowanie udanych rozwiązań QSEE Superlite z jednoczesną poprawą interfejsu, miało to na celu przyspieszenie oraz ułatwienie procesu tworzenia modelu przy pomocy diagramów przepływu danych. Elementy DFD wyglądem odpowiadają elementom zawartym w pierwszej aplikacji, różni się natomiast sposób umieszczania ich na płótnie. Umieszczenie zachodzi po przeciągnięciu elementu z paska bocznego, usuwa to konieczność każdorazowego klikania, co przy dużej liczbie elementów może skrócić czas potrzebny do tworzenia diagramów przepływu danych.

4. Metodyka badawcza

Celem artykułu było przeprowadzenie analizy możliwości wykorzystania aplikacji do tworzenia diagramów przepływu danych. Badanie przeprowadzono na dwóch aplikacjach QSEE Superlite oraz DFD Maker, na grupie dwudziestu uczestników. Uczestnikami badania byli absolwenci Politechniki Lubelskiej z kierunku informatyka. Znajomości podstaw diagramów DFD, a także obycie z aplikacjami służącymi do tworzenia owych diagramów, pozwoliła na dokładniejsze sprawdzenie aplikacji. Badanie podzielono na dwa scenariusze badawcze. Przed rozpoczęciem badania na podstawie opracowanych scenariuszy badawczych, każdy użytkownik otrzymywał czas, aby zapoznać się z działaniem sprawdzanej aplikacji. Podczas zapoznawania się osoby biorące udział w badaniu tworzyły proste diagramy.

Pierwszy scenariusz badawczy wymagał od użytkowników odwzorowania diagramu przepływu danych, przedstawionego na rysunku 1. Miało to na celu sprawdzenie szybkości tworzenia diagramów DFD w danej aplikacji. Zebrane na tej podstawie czasy posłużyły do obliczenia średniej oraz odchylenia standardowego dla każdej z aplikacji. Ze względu na czas potrzebny na zapoznanie się z każdą aplikacją odwzorowywany model jest prosty, co miało na celu skrócenie badań.

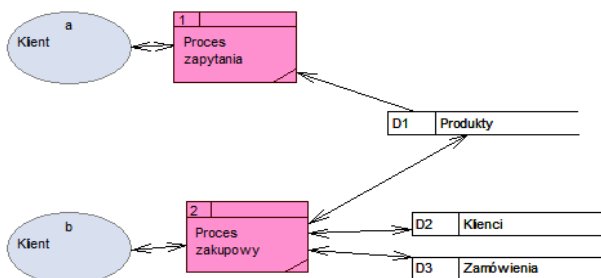


Rysunek 1: Odwzorowywany diagram przepływu danych.

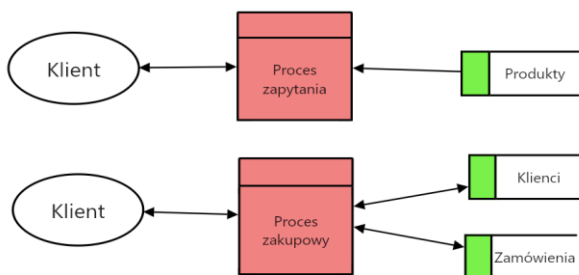
Drugi scenariusz badawczy pozwalał użytkownikom na stworzenie dowolnie wielu diagramów przepływu danych. Po zakończeniu całego procesu, użytkownicy ocenili przyjazność interfejsu, a także łatwość tworzenia DFD. Jak w poprzednim scenariuszu, tu również na podstawie zebranych wyników obliczono średnią, a także odchylenie standardowe dla każdej z aplikacji. Użytkownicy oceniali aplikacje w skali 1 do 10, gdzie 1 oznacza, iż aplikacja nie nadaje się do tworzenia diagramów, natomiast 10 oznacza uznanie aplikacji, za perfekcyjne narzędzie do tworzenia DFD.

5. Wyniki badań

Pierwszym badanym aspektem był czas potrzebny do odwzorowania diagramu przepływu danych. Wyniki dla średniego czasu oraz odchylenia standardowego zaprezentowano w Tabeli 1. Rysunek 2 oraz 3 obrazuje przykładowe odwzorowanie diagramu przepływu danych w każdej z aplikacji.



Rysunek 2: Diagram stworzony w aplikacji QSEE.

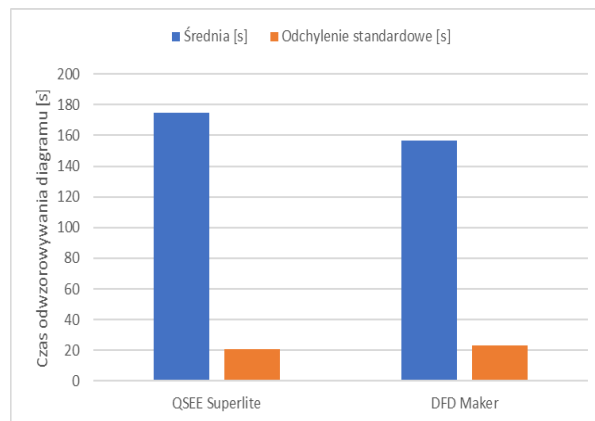


Rysunek 3: Diagram stworzony w aplikacji DFD Maker.

Tabela 1: Średni czas odwzorowywania diagramu przepływu danych przez badanych

Liczba prób: 20	Średni czas tworzenia [s] ± odchylenie standardowe
QSEE	174,7 ± 20,82
DFD Maker	156,6 ± 22,897

Zestawione wyniki dla pierwszego scenariusza badawczego przedstawiono również w formie wykresu na Rysunku 4. Przedstawiają one w sekundach średni czas potrzebny do odwzorowania diagramu przepływu danych.



Rysunek 4: Test średniego czasu potrzebnego do odwzorowania diagramu.

Drugim badanym aspektem była przejrzystość interfejsu, a także łatwość tworzenia diagramów przepływu danych. Wyniki przedstawiono w Tabeli 2 oraz 3.

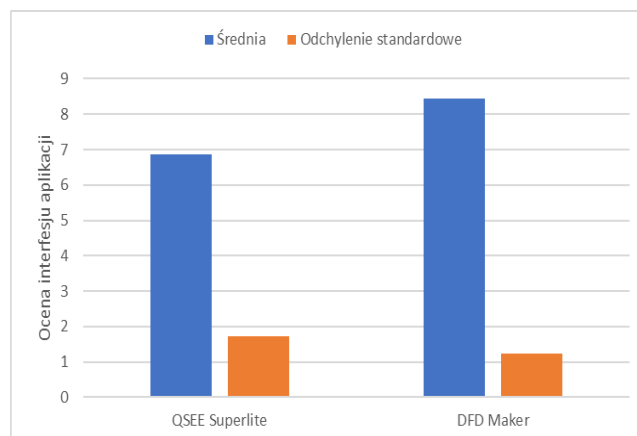
Tabela 2: Średnia ocena interfejsu aplikacji przez badanych

Liczba prób: 20	Średnia ocena aplikacji ± odchylenie standardowe
QSEE	6,85 ± 1,725
DFD Maker	8,45 ± 1,234

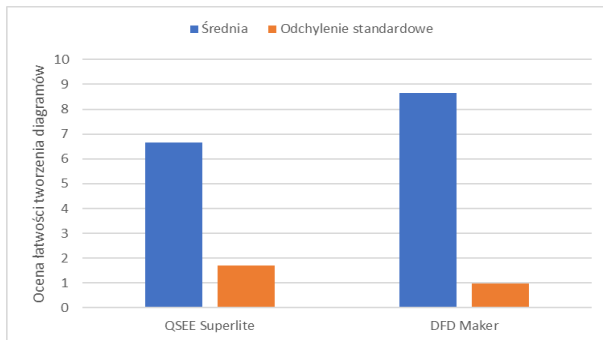
Tabela 3: Średnia ocena łatwości tworzenia diagramów

Liczba prób: 20	Średnia ocena aplikacji ± odchylenie standardowe
QSEE	6,65 ± 1,694
DFD Maker	8,65 ± 0,988

Zestawione wyniki dla drugiego scenariusza badawczego przedstawiono na Rysunkach 5 oraz 6.



Rysunek 5: Średnia ocena przejrzystości interfejsu przez użytkowników.



Rysunek 6: Średnia ocena łatwości tworzenia DFD.

6. Wnioski

W niniejszym artykule przedstawiono analizę możliwości wykorzystania aplikacji do tworzenia diagramów przepływu danych, a także omówiono zalety jakie płyną ze stosowania owych diagramów w projekcie systemu informatycznego. Wybrano dwie aplikacje do porównania, pierwszą z nich jest QSEE Superlite, drugą jest stworzona na potrzeby badania we Flutterze.

Badania, które zostały wykonane polegały na zmierzeniu czasu potrzebnego do odwzorowania diagramu przepływu danych, a także do zebrania ocen użytkowników na temat przejrzystości interfejsu oraz łatwości tworzenia DFD w każdej z aplikacji.

W przypadku badań pomiaru czasu, wyniki przedstawiają, iż DFD Maker pozwala na szybsze tworzenie diagramów, pomimo iż to QSEE Superlite było drugą aplikacją testowaną przez każdego użytkownika, co mogło pomóc QSEE, gdyż ponowne odtworzenie modelu jest szybsze. Pomimo tego dla zaproponowanego DFD różnica średnich czasów wynosi 18 sekund, co przy skomplikowanych modelach, jak i konieczności tworzenia wielu diagramów daje dużą oszczędność czasu.

Badania nad przejrzystością interfejsu, a także łatwością tworzenia diagramów przepływu danych dostarczają wyników, które obrazują, iż DFD Maker posiada przejrzystszy, a co za tym idzie przyjaźniejszy interfejs. Badani zostali poproszeni o ocenienie aplikacji w skali od 1 do 10, gdzie 1 oznaczało niemożliwość współpracy z narzędziem, a 10 stwierdzało, iż aplikacja jest idealnym narzędziem. Użytkownicy oceniali średnio interfejs DFD Makera wyżej o 1,6 punktu, przy średniej wynoszącej 8,45, natomiast QSEE Superlite osiągnęło średnią 6,85. Sama średnia ocena łatwości tworzenia diagramów również wypada korzystniej dla aplikacji opracowanej przez autora względem QSEE Superlite i wynosi 2 punkty. QSEE osiągnęło średnią 6,65 natomiast DFD Maker 8,65.

Literatura

- [1] S. Adi, D. M. Kristin, Strukturisasi Entity Relationship Diagram dan Data Flow Diagram Berbasis Business Event-Driven, ComTech: Computer, Mathematics and Engineering Applications 5(1) (2014) 26-34 <https://journal.binus.ac.id/index.php/comtech/article/view/12577>.
- [2] M. Nejad-Sattary, An Extended Data Flow Diagram Notation for Specification of Real-Time Systems 1990 <https://openaccess.city.ac.uk/id/eprint/28541/>.
- [3] R. Ibrahim, S. Y. Yen, Formalization of the data flow diagram rules for consistency check, International Journal of Software Engineering & Applications 1(4) (2010) 95-111 <https://arxiv.org/ftp/arxiv/papers/1011/1011.0278.pdf>.
- [4] M. Odlanicka-Poczobutt, E. Kulińska, Zastosowanie diagramu przepływu danych w analizie systemowej procesu realizacji zamówień klienta interco 1 (2015) 936-948 http://repolis.bg.polsl.pl/Content/36138/Zastosowanie%20diagramu_97076.pdf.
- [5] A. Liaushuk, A. Jasiak, Method of organizing of the data flow in business entity on the basis of data flow diagram and the theory of queues, Research in Logistics & Production 9(1) (2019) 31-38 <https://yadda.icm.edu.pl/baztech/element/bwmeta1.element/baztech-93d60794-6b31-45aa-9af2-e8f2957aecb8>.
- [6] Uniwersytet Kapsztadzki w Południowej Afryce [07.06.2023] https://www.cs.uct.ac.za/mit_notes/software/htmls/ch06.html.
- [7] J. R. Warren, J. W. Stott, A. F. Norcio, Stochastic simulation of information systems designs from data flow diagrams, Journal of Systems and Software 18(2) (1992) 191-199 <https://www.sciencedirect.com/science/article/abs/pii/0164121292901276>.
- [8] H. Alshareef, K. Tuma, S. Stucki, G. Schneider, R. Scandariato, Precise Analysis of Purpose Limitation in Data Flow Diagrams, Proceedings of the 17th International Conference on Availability, Reliability and Security (2022) 1-11 <https://dl.acm.org/doi/abs/10.1145/3538969.3539010>.
- [9] P. D. Bruza, T. Van der Weide, The Semantics of Data Flow Diagrams, Nijmegen TR: University of Nijmegen, Department of Informatics, Faculty of Mathematics and Informatics 1989 <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=70cfc87f97d9aacdcac857f7222086081d965cd3>.
- [10] W. Wulandari A. D. Y. Widiatoro, Design Data Flow Diagram for Supporting the User Experience in Applications, Design Data Flow Diagram for Supporting the User Experience in Applications 25(2) (2017) 14-20 <http://repository.unika.ac.id/17139/>.
- [11] R. Ganesh, G. Prabu, Determination of Internet Banking Usage and Purpose with Explanation of Data Flow Diagram and Use Case Diagram, International Journal of Management and Humanities 4(7) (2020) 52-58 <https://www.ijmh.org/wp-content/uploads/papers/v4i7/G0674034720.pdf>.
- [12] S. Zindl, Flutter on Windows Desktop: a use case based study, Bachelor's thesis [07.06.2023] <https://elib.uni-stuttgart.de/handle/11682/11740>