**PAŃSTWOWA WYŻSZA SZKOŁA ZAWODOWA**

**W NOWYM SĄCZU**

**INSTYTUT TECHNICZNY**

**PRACA DYPLOMOWA**

**Społecznościowy agregat wiadomości z dedykowanymi filtrami artykułów i mechanizmem współdzielenia**

**Autor: Kmak Karol**

**Kierunek: Informatyka stosowana**

**Nr albumu: 28264**

**Promotor: dr hab. Witold Przygoda**

**Akceptacja promotora: ……………………………………………………**  data i podpis

**NOWY SĄCZ 2022**

Spis treści

[Wstęp 4](#_Toc93442477)

[1 Cel i założenia projektu 5](#_Toc93442478)

[1.1 Środowiska programowe 6](#_Toc93442479)

[2 Implementacja 7](#_Toc93442480)

[2.1 Aplikacja mobilna 7](#_Toc93442481)

[2.2 System bazodanowy 8](#_Toc93442482)

[2.3 Strona interntowa do zarządzania agregatami i kod serwera 12](#_Toc93442483)

[3 Testy aplikacji 12](#_Toc93442484)

[3.1 Testy aplikacji mobilnej 12](#_Toc93442485)

[3.2 Testy systemu bazodanowego i narzędzi do zarządzania agregatami 12](#_Toc93442486)

[4 Podsumowanie i wnioski 13](#_Toc93442487)

[5 Bibliografia 14](#_Toc93442488)

[6 Spis rysunków 14](#_Toc93442489)

[7 Załączniki 14](#_Toc93442490)

# Wstęp

Rozwój internetu umożliwiający co raz większej liczbie osób na realizację swoich dziennikarskich pasji spowodował powstanie niezliczonej liczby portali, blogów i innych źródeł, które dla użytkownika nie chcącego poświęcać zbyt wielu godzin na śledzenie różnych portali by móc być na bieżąco z wydarzeniami ze świata stworzył rynek agregatów wiadomości. Są to usługi zbierające artykuły z wielu portali i przedstawiające wybrane artykuły według zainteresowań użytkownika usługi.

Istniejące rozwiązania działają na podstawie jednej z dwóch podstawowych metod. Pierwsza metoda to wykorzystanie algorytmów, które na podstawie zbieranych danych o ilości odwiedzin danej strony, czasie spędzonym na stronie przekazują użytkownikowi artykuły, które są w danym momencie ważne, interesujące i jak najlepiej dopasowane do użytkownika dzięki porównaniu z historycznymi danymi zainteresowań użytkownika. Rozwiązanie to potrafi najczęściej trafnie określić zbiór artykułów mogących zainteresować czytelnika, ale nie jest w stanie rozróżnić informacji nieprawdziwych, czy tzw. clickbaitów i innych artykułów nie przedstawiający pożądanych dla użytkownika wartości.

Drugim z popularnych rozwiązań są artykuły specjalnie wyselekcjonowane przez zajmującą się tym redakcję. Zapewnia to o wiele lepszą jakość artykuł dzięki weryfikacji przez osoby doświadczone, ale wadą tego rozwiązania jest fakt, że pracownicy takiej redakcji nie są w stanie przeanalizować wszelkich możliwych źródeł, oraz mają swoje poglądy, które mogą wpłynąć na proces decyzyjny wybrania, bądź odrzucenia danego artykułu.

Dlatego platforma pozwalająca na wykorzystanie najlepszych cech dominujących rozwiązań, a jednocześnie minimalizująca ich wady jest rozwiązaniem mogącym łatwo zdobyć pozycję na rynku pozwalającą na stabilne finansowanie przynosząc zyski dla właściciela platformy. Społecznościowy agregat wiadomości ma za zadanie oddać poszczególnym użytkownikom, organizacjom możliwość pełnienia funkcji redakcji filtrującej treści dla innych użytkowników. Użytkownik aplikacji będzie miał możliwość wybrania treści zweryfikowanych przez konkretnego, wybranego przez siebie innego użytkownika i/lub za pomocą systemu tagów wybrać zakres zainteresowań, na podstawie którego będą mu przekazywane subskrybowane przez niego treści. W celu zapewnienia użytkownikowi możliwości odfiltrowania treści dla niego mało ważnych, takich jak np. plotki aplikacja posiadać będzie możliwość określenia kilkustopniowego poziomu ważności.

# Cel i założenia projektu

Celem pracy jest stworzenie aplikacji mobilnej systemu Android pozwalającej użytkownikowi przeglądanie listy polecanych artykułów, która będzie pobierana i aktualizowana z serwera bazodanowego. Aplikacja po naciśnięciu przez użytkownika w artykuł otworzy okno domyślnej przeglądarki zainstalowanej na urządzeniu i załaduje wybrany przez użytkownika artykuł. Użytkownik ze specjalnej zakładki będzie mógł przeglądać dostępne agregaty i je zasubskrybować dzięki czemu artykuły z tego agregatu będą się pojawiać w głównej zakładce aplikacji. Ponadto użytkownik będzie mógł śledzić wiadomości bez konieczności subskrypcji żadnego z agregatów dzięki systemowi tagów przy wykorzystaniu którego będzie mógł np. śledzić artykułu dotyczącego wydarzeń z wybranej przez siebie miejscowości, lub na podstawie innych, wybranych przez niego tagów. Aplikacje będzie też oferować możliwość odfiltrowania z listy wyników artykułów nieistotnych za pomocą kilkustopniowej własności ważność każdego artykułu, która będzie ustalana przez zarządzających poszczególnymi agregatami, a w przypadku niesubskrybowania żadnego z agregatów na podstawie średniej ważności przydzielonej do artykułu. W przypadku gdy użytkownik aplikacji nie będzie miał czasu by zapoznać się z treścią artykułu, lub gdy z innych powodów będzie chciał zapisać artykuł do przeczytania w innym terminie otrzymaną taką funkcję, oraz specjalną zakładkę przechowującą zapisane przez użytkownika artykuły.

Drugim celem pracy jest stworzenie lokalnego systemu bazodanowego przechowującego informacje o agregatach i strony internetowej pozwalającej użytkownikom na zarządzanie swoimi agregatami poprzez dodawanie nowych artykułów, zmianę ich właściwości lub usunięcie ich ze swojego agregatu.

# Środowiska programowe

Android Studio – oficjalne otwarto źródłowe środowisko programistyczne dla mobilnych systemów operacyjnych Android. Środowisko oferuje obsługę języków programowania takich jak m.in. Java, C++, Kotlin. Zawiera funkcjonalności takie jak budowanie interfejsu aplikacji za pomocą narzędzia graficznego, zintegrowany emulator pozwalający przetestować działanie aplikacji z poziomu komputera, oraz narzędzia do kontroli kompatybilności aplikacji z poszczególnymi wersjami systemu Android, a także funkcję automatycznego tłumaczenia kodu języka Java na kod w języku Kotlin. Aplikacja będzie pisana w językach Kotlin oraz Java, a do tworzenia interfejsu graficznego wykorzystane będzie narzędzie do graficznego generowania interfejsu oraz język znaczników XML.

Funkcje lokalnego serwera będą tworzone za pomocą USBWebserver programu pozwalającego na uruchomienie serwera Apache, MySQL, PHP, oraz phpMyAdmin. Pozwala on na bezproblemowe przeniesie i uruchomienie na dowolnym komputerze przez zastosowanie np. pamięci USB bez konieczności instalacji na komputerze z którego jest używany. Program ten został wykorzystany na podstawie łatwości użytkownika, oraz swojej możliwości pozwalającej na szybkie udostępnienie usługi serwera z poziomu dowolnego komputera bez konieczności posiadania uprawnień administratora na tym komputerze.

Aplikacja bazodanowa zostanie stworzona przy użyciu phpMyAdmin/MySQL będącego narzędziem służącym do tworzenia i zarządzania relacyjnymi bazami danych MySQL, oferuje on zarządzanie bazą danych przy użyciu poleceń tekstowych, a także interfejsu graficznego dostępnego z poziomu przeglądarki internetowej.

Strona internetowa zawierająca narzędzia dla zarządzających agregatami zostanie wykonana z pomocą serwera Apache i przy wykorzystaniu języka znaczników HTML, oraz skryptowego języka programowania JavaScript. Obsługa zapytań aplikacji, oraz strony internetowej wykonana będzie z wykorzystaniem języka PHP.

# Implementacja

dfg

# Aplikacja mobilna

hgfh

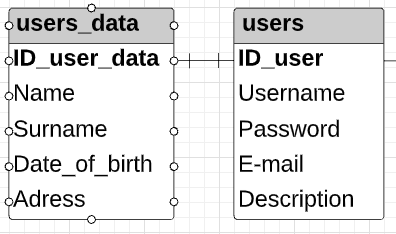
# System bazodanowy

Baza danych korzystać będzie z ustandaryzowanego systemu nazewnictwa. Wszelkie nazwy zapisane będą w języku angielskim, który jest standardowym sposobem zapisu w informatyce. Nazwy tabel zapisywane małymi literami w liczbie mnogiej. Nazwy każdej kolumny zaczynać się będą z dużej litery i będą zapisywane w liczbie pojedynczej, w przypadku wystąpienia więcej niż jednego słowa oddzielane one będą znakiem „\_”. Klucz główny będzie zapisany jako ID\_nazwa\_tabeli, gdzie nazwa będzie zapisaną w liczbie pojedynczej nazwą tabeli do której należy. Klucz obcy zapisywany będzie w analogicznej formie, zmienionym o kolejność zapisu słów tak, że nazwa tabeli będzie zapisana jako pierwsza i po niej będzie dodany człon „ID:. Nazwy tabel łączących będą zapisywane stosując nazwy łączonych tabel oddzielając je znakiem „-”, lub inaczej gdy tabela będzie przechowywać dane inne niż te niezbędne do jej funkcjonowania i znacząco rozbudowujące jej funkcjonalność, klucz główny tabeli będzie zapisywany stosując człon „ID\_”, oraz dwie pierwsze litery łączonych tabel oddzielonych znakiem „-”.

## Tabela użytkownicy

Aplikacja w bazie danych będzie przechowywać numer ID użytkownika, nadawany automatycznie w celu identyfikacji przez system, jego nazwę wymyśloną i wybraną przez użytkownika w momencie rejestracji, nazwa musi być unikalna i nie będzie możliwości jej zmiany po zarejestrowaniu konta, adres e-mail w celu zapewnienia możliwości odzyskania hasła, oraz krótki opis. Opis ten jest sposobem by przekazać przez użytkownika krótką notatkę o sobie i może być edytowany w każdej chwili. Pola nazwy użytkownika, hasła oraz adresu e-mail są obowiązkowe. Hasło przechowywane jest w bazie w formie zaszyfrowanego tekstu.

Dodatkowo w przypadku głównego zarządzającego agregatem obowiązkowym jest uzupełnienie danych osobistych, czyli imienia, nazwiska, daty urodzenia, oraz adresu zamieszkania przechowywanych w osobnej tabeli, połączonej tak jak na rysunku poniżej. W przypadku kolumny data zrezygnowano z zasady atomowości danych, w celu uproszczenia struktury bazy danych. Zakładana rzadkość wykorzystywania tych danych nie uzasadnia zwiększenia skomplikowania bazy.

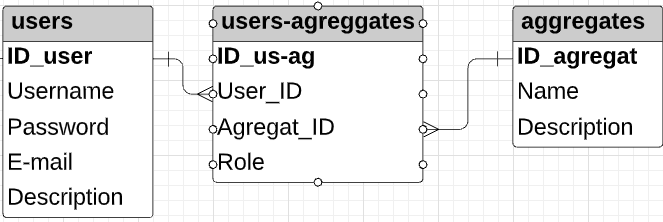


Rysunek 1. Tabele przechowujące dane użytkownika

## Połączenia użytkowników z zarządzanymi agregatami

Aplikacja powinna umożliwiać zarządzanie poszczególnymi agregatami dla wielu użytkowników, aby zapewnić administratorowi możliwość łatwego i przejrzystego podglądu historii zmian dokonywanych przez każdego z użytkowników mających uprawnienia do zarządzania agregatem. A z racji faktu, że użytkownik może wyrażać chęć zarządzania więcej niż jednym agregatem np. w sytuacji gdy organizacja do której należy podjęła decyzję by tworzyć więcej niż jeden agregat aby zapewnić swoim użytkownikom jak najlepsze dopasowanie do ich preferencji. Z tego powodu koniecznym jest umieszczenie tabeli łączącej, która przechowywać będzie klucze identyfikujące z tabel przechowujących dane użytkowników i agregatów. Dzięki zastosowaniu tabeli łączącej dowolny użytkownik będzie mógł być połączony z dowolną liczbą agregatów, a jednocześnie każdy z tych agregatów będzie mógł być połączony z dowolną liczbą zarządzających nim użytkowników. Połączenie tabel wykonane jest jak na rysunku poniżej.

W tabeli łączącej zawiera się też kolumna definiująca uprawnienia użytkownika do wprowadzania zmian w agregacie. Uprawnienia są zapisywane i odczytywane w sposób binarny, a przechowywane w bazie po zamianie tej liczby na system dziesiętny. Liczba jeden oznacza posiadanie danego uprawnienia, a zero jego brak. Uprawnienia licząc od pozycji najmniejszej do największej to: główny administrator (może być tylko jeden dla danego agregatu), usuwanie administratorów agregatu, zmiana ról, dodawanie nowych administratorów, usuwanie artykułów, wprowadzanie zmian w artykułach, dodawanie artykułów.

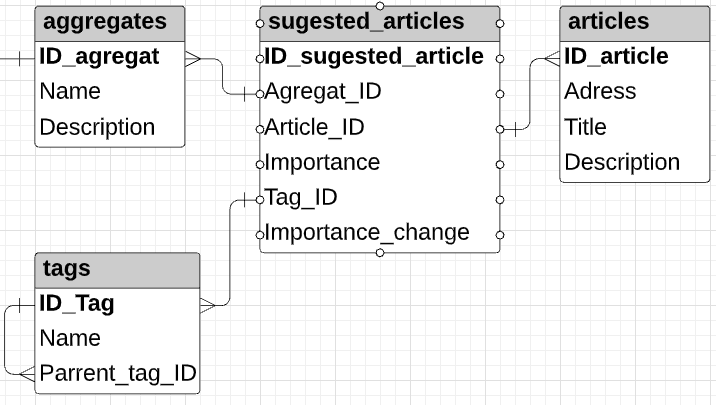


Rysunek 2. Tabele obsługujące połączenie użytkownika z agregatem

## Główna część działania agregatu

Zarządzanie agregatem będzie realizowane za pomocą tabeli „aggregates” przechowującej nazwę agregatu, oraz jego krótki opis, tabeli „articles” przechowującej tytuł artykułu, jego adres i pierwsze słowa artykułu mające zachęcić użytkownika do jego przeczytania. Tabele połączone będą tabelą łączącą „sugested\_articles” (rysunek poniżej) zawierającą oprócz kluczy głównych łączonych tabel kolumnę określającą ważność za pomocą wartości liczbowej w zakresie 1-6, gdzie jeden oznacza artykuły o najmniejszej ważności takie jak np. plotki, a sześć oznacza artykuł bardzo ważny. Tabela będzie też zawierać kategorię artykułu, oraz zmienną określającą zmianę ważności artykuły po przejściu do bardziej ogólnej kategorii. W tabeli „tags” zawarta będzie nazwa kategorii artykułu, oraz klucz przechowujący dane o kategorii bezpośrednio wyżej, bardziej ogólnej.

Kolumna „Importance\_change” wskazuje na zmianę wartości ważności artykułu w czasie zmiany kategorii na nadrzędną. Przy pierwszym przejściu ważność jest zmniejszana o wskazaną wartość, a podczas każdego następnego przejścia zmienna ta jest podwajana, osiągając tym samym coraz większy spadek wartości wraz ze zmianą kategorii na bardziej ogólną.



Rysunek 3. Zarządzanie agregatem

# Strona interntowa do zarządzania agregatami i kod serwera

fghfg

# Testy aplikacji

hfdhf

# Testy aplikacji mobilnej

Gfhfhdfhd

# Testy systemu bazodanowego i narzędzi do zarządzania agregatami

gdfghdfgh

# Podsumowanie i wnioski

Rozdział ten powinien stanowić zwieńczenie pracy. Wyraźnie powinno być   
w nim pokazane zrealizowanie zapowiedzianego celu i zakresu pracy projektu lub zadania badawczego.

Najogólniej, należy tu zwięźle przedstawić, co zrobiono w danej pracy dyplomowej oraz podać syntezę uzyskanych wyników. Ta część pracy powinna wykazać, że cel sformułowany we wstępie został spełniony.

Należy przyjąć zasadę że wszystkie stwierdzenia oraz wnioski szczegółowe   
i uogólnienia muszą wynikać bezpośrednio z treści pracy. Główną treścią wniosków powinna być synteza wyników szczegółowych. Celowym jest zapisanie stwierdzeń o charakterze metodycznym, odnoszących się np. do dostępności danych źródłowych lub przydatności stosowanych metod doświadczalnych czy obliczeniowych.

Pożądane jest przedstawienie wniosków o charakterze praktycznym, które mogą mieć postać zaleceń adresowanych do konkretnych rozwiązań technicznych. Celowe są również wnioski o charakterze subiektywnym, np. odnoszące się do umiejętności i konieczności samodzielnego pozyskiwania i przetwarzania – na potrzeby zrealizowanej pracy dyplomowej – wiedzy z obszarów nie objętych programem kształcenia na ukończonych studiach.

Syntetyczne i krytyczne podsumowanie uzyskanych wyników (z podkreśleniem wkładu własnego autorki/autora pracy), a także **wytyczne do ewentualnych dalszych działań** (wykraczających poza zakres pracy dyplomowej

Aplikacja może zostać ulepszona poprzez dodanie użytkownikowi opcji zarządzania polecanymi artykułami z poziomu aplikacji mobilnej, a także tworząc rozszerzenie do przeglądarek internetowych pozwalające na dodanie artykuły bez konieczności logowania się do strony pełniącej funkcję centrum zarządzania agregatem tworzonym przez użytkownika. Dodatkowo pożądanym rozwiązaniem byłoby dodanie obsługi wielu kont użytkowników zarządzających jednym agregatem, wraz z stroną pozwalającą na śledzenie zmian wprowadzanych przez użytkowników. Dzięki temu większe organizacje miałyby narzędzie pozwalające na ocenę jakości pracy poszczególnych użytkowników. Kolejną zmianą, kluczową by aplikacja mogła doczekać się udostępnienia użytkownikom jest implementacja oprogramowania serwerowego na ogólnodostępnym serwerze.

# Bibliografia

Alifio Tutorials(11.02.2018). *How to connect android application to MySQL Database in XAMP Server*. Pobrane z https://youtu.be/4e8be8xseqE, data pobrania [12.02.2022]

Anupam Chugh(16.06.2020). *Android Alert Dialog using Kotlin*. Pobrane z https://www.journaldev.com/, data pobrania [12.02.2022]

Dawn Griffiths, David Griffiths(2020). *Kotlin. Rusz głową!* O'Reilly/Helion

# Spis **rysunków**

Rysunek 1. Tabele przechowujące dane użytkownika 9

Rysunek 2. Tabele obsługujące połączenie użytkownika z agregatem 10

Rysunek 3. Zarządzanie agregatem 11

# Załączniki