Zielona Góra 14.06.2020

Przechowywanie i Przetwarzanie Danych

Maciej Sabik,Bartosz Musielak, Szymon Stachów, Mateusz Żmuda

1. Cel i założenia projektu

Zadaniem zespołu było stworzenie bazy danych oraz przechowywanie w niej danych.

2. Opis istniejących rozwiązań

W tym celu wykorzystana została biblioteka Room, dzięki której możliwe było przechowywanie danych. Dzięki czemu praca z obiektami SQLiteDatabase była łatwiejsza, jak i również pomogło to zoptymalizować niepotrzebną ilość kodu oraz korzystanie z dodatkowych klas POJO. Wykorzystując bibliotekę Rooma ułatwiona była weryfikacja zapytań SQL w czasie kompilacji, dzięki czemu w procesie kompilacji zostanie dostarczona nam informacja o błędnym sformułowaniu zapytania.Kolejnym komponentem wykorzystanym do tworzenia tabel było Entity. Dzięki czemu każda klasa, która posiadała adnotację @Entity, tworzyła tabele, w których pola jakie znajdowały się w danej klasie, wskazywały kolumny które zostały utworzone w danej tabeli. Następnym rozwiązaniem był interfejs Dao, który jest odpowiedzialny za definiowanie metody dostępu do bazy danych.

2.1.Opis najczęściej występujących problemów

Głównym problemem w bibliotece rooma był brak możliwości wykorzystać tej samej bazy danych na kilku aktywnościach, ponieważ za każdym razem tworzyła się nowa instancja tej samej bazy danych. Remedium na ten problem było wykorzystanie nowej biblioteki o nazwie

2.2.Opis zastosowanych technologii

Dagger2 - umożliwia wstrzykiwanie zależności w danej aktywności, dzięki temu można było wstrzyknąć w wykorzystywane aktywności wcześniej już utworzoną instancję bazy danych.

3. Opis występujących problemów w sprawach organizacyjnych

Ze względu na zaistniałą sytuację, grupa miała problem z komunikacją, oraz opisaniem zaistniałego problemu, rozwiązaniem była komunikacja z wykorzystaniem programu Discord, oraz przesyłanie stworzonych projektów poprzez GitHub, co usprawniło pracę w zespole, dzięki czemu problemy mogły zostać omówione jak i również zniwelowane.

3.1. Opis rozwiązań organizacyjnych

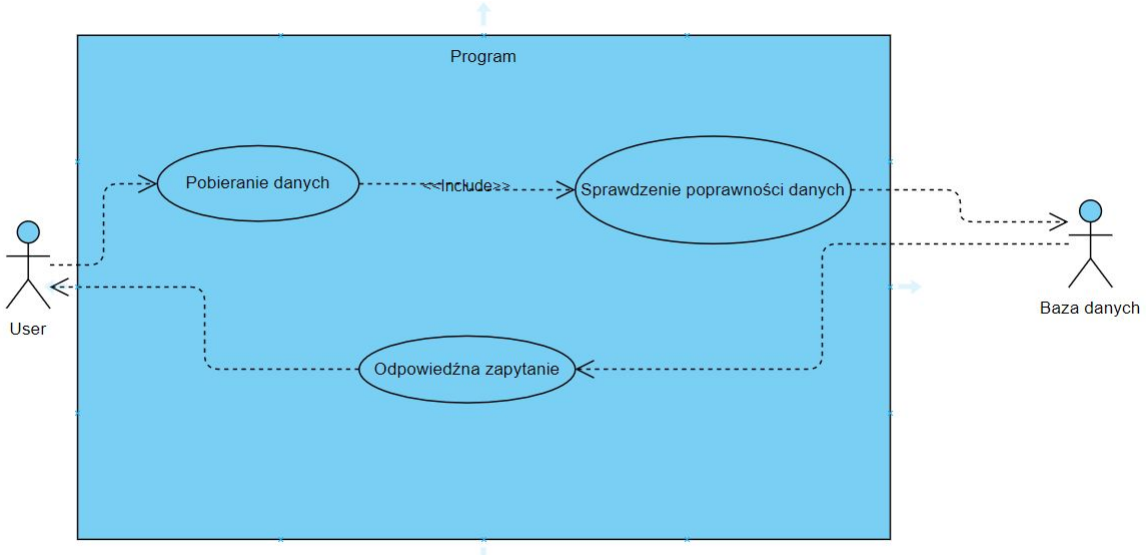
Komunikacja w grupie, dzięki selektywnie dobranym osobą, była uproszczona oraz ułatwiała dobór odpowiednich programów jak i wykorzystanych bibliotek, w celu zneutralizowania przyszłych problemów ze zgodnością plików.

3.2. Opis rozwiązań technologicznych

Do komunikacji wykorzystany został program Discord, dzięki czemu na bieżąco mogliśmy komunikować się ze sobą, jak i ukazywać rezultaty swojej pracy, aby mieć wgląd do stworzonych programów, stworzone zostało repozytorium na GitHubie, co spowodowało, że pliki udostępnione przez inne osoby mogły bez problemu zostać użyte w programach innych członków zespołu.

4. Diagram użycia Use-Case Diagram

Poniżej został przedstawiony Use-Case Diagram dotyczący modułu bazy danych.



Rys.1

5. Tabele przypadków użycia Use-Case Templates

Poniżej został przedstawiony Use-Case Templates dotyczący modułu bazy danych i aktywności jakie może wykonać użytkownik na bazie danych poprzez problem a także korelację pomiędzy bazą a użytkownikiem.

	Pobieranie danych
Aktorzy	Użytkownik

Warunki początkowe	● Wysłane zapytanie do bazy danych
Zdarzenie inicjujące	● Użytkownik klika w guzik odpowiedzialny za pobieranie danych
Przebieg w krokach	1. Użytkownik klika guzik do pobrania danych 2. System wyświetla dane takie jak: napięcie na baterii, temperaturę, wilgotność powietrza 3. Dane zostają zapisane do bazy danych 4. System wyświetla potwierdzenie zapisania danych
Sytuacje wyjątkowe	● Użytkownik znajduje się poza zasięgiem czujnika co skutkuje, brakiem nowych danych. System wyświetla stosowny komunikat informujący użytkownika o zbyt dużej odległości i uniemożliwia pobranie danych dopóki nie znajdzie się w zasięgu czujnika.

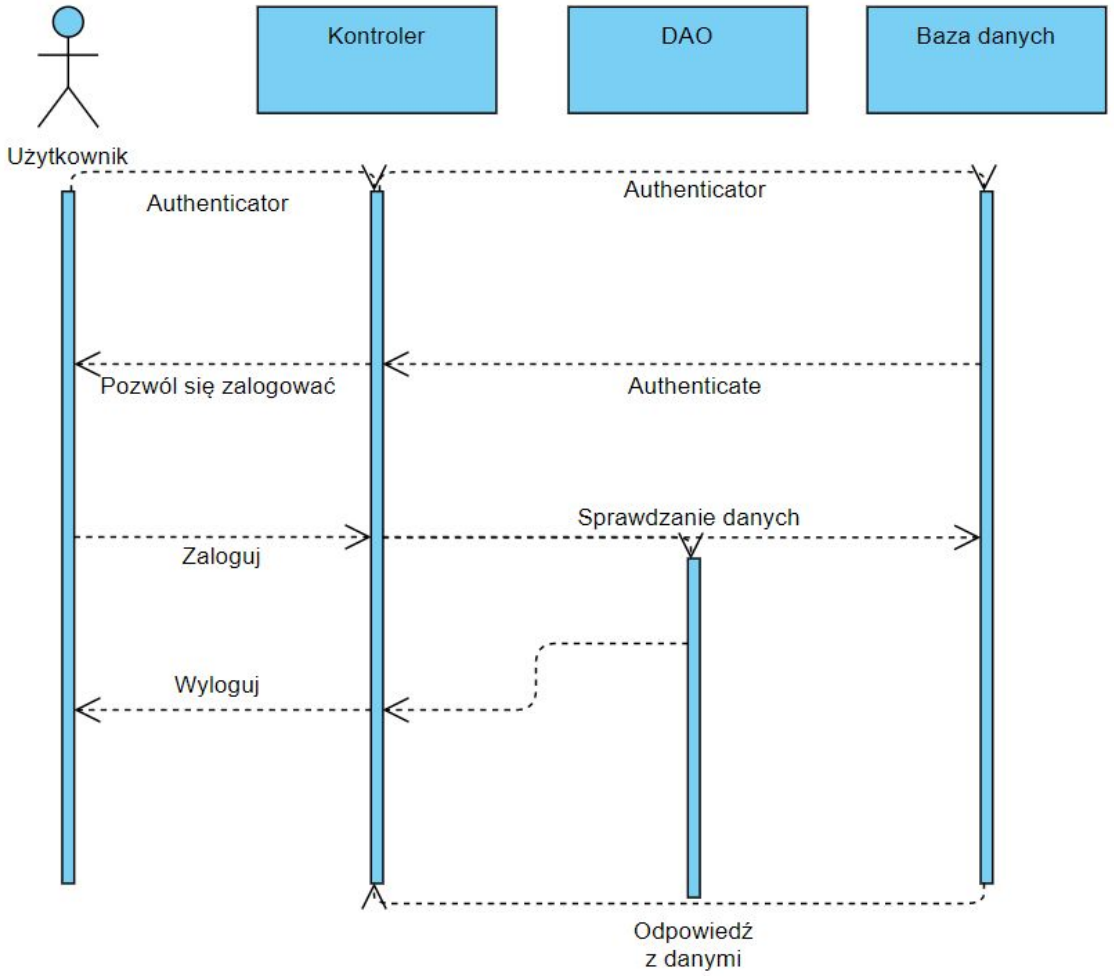
	Odpowiedź na zapytanie
Aktorzy	Użytkownik
Warunki początkowe	● Użytkownik posiada zapisane dane w aplikacji ● Użytkownik przeszukuje aplikację w poszukiwaniu danych

Zdarzenie inicjujące	● Użytkownik klika w opcję do wyświetlania danych
Przebieg w krokach	1. Użytkownik klika w guzik do wyświetlenia danych 2. Użytkownik dostaje dane w postaci tabeli
Sytuacje wyjątkowe	● Brak danych w bazie danych

	Sprawdzanie poprawności danych
Aktorzy	Użytkownik
Warunki początkowe	● Użytkownik jest zalogowany ● Użytkownik pobiera dane
Zdarzenie inicjujące	● Użytkownik w opcję do pobierania danych
Przebieg w krokach	1. Użytkownik klika guzik do pobierania danych 2. System sprawdza poprawność danych 3. System zapisuje dane do bazy danych
Sytuacje wyjątkowe	● Dane nie pobrały się prawidłowo lub pobrane dane są niepełne

6. Diagram przepływu sterowania Control-Flow Diagram

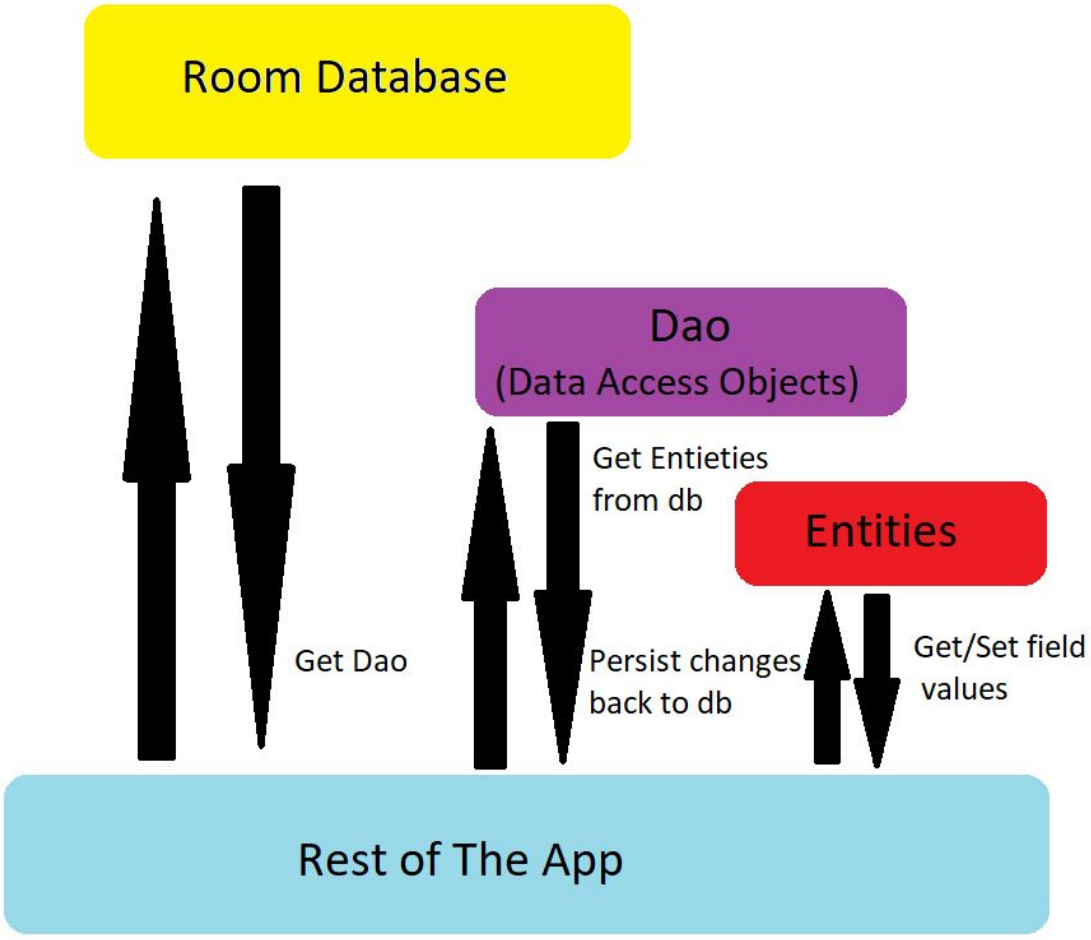
Poniżej został przedstawiony Control-Flow Diagram



7. Usługi oraz funkcje

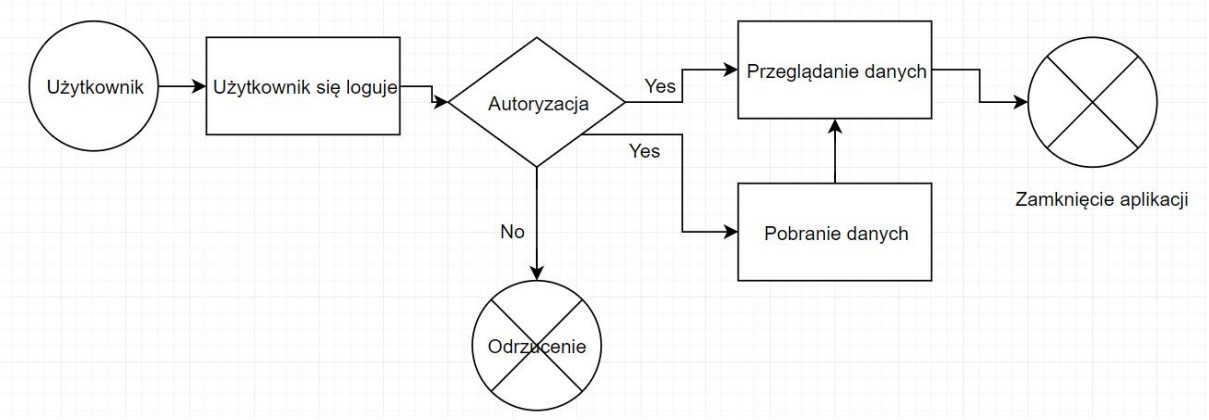
Użyto funkcji CRUD, która służy jest stosowany do tworzenia lub dodania nowych informacji (create), odczytania lub wyświetlenia istniejących informacji (read), modyfikowania lub edycji istniejących informacji (update) lub też usuwania istniejących informacji (delete). Dzięki czemu zarządzanie danymi w bazie danych jest intuicyjne i proste. Funkcja CRUD jest wykorzystana w prawie każdym wygenerowanym automatycznie kontrolerze, dzięki czemu wyręcza programistę w pisaniu zapytań SQL.

8. Diagram komponentów/obiektów



9. Diagram funkcjonalny oraz strukturalny

Poniżej został opisany diagram funkcjonalny dla modułu bazy danych



10. Harmonogram projektu

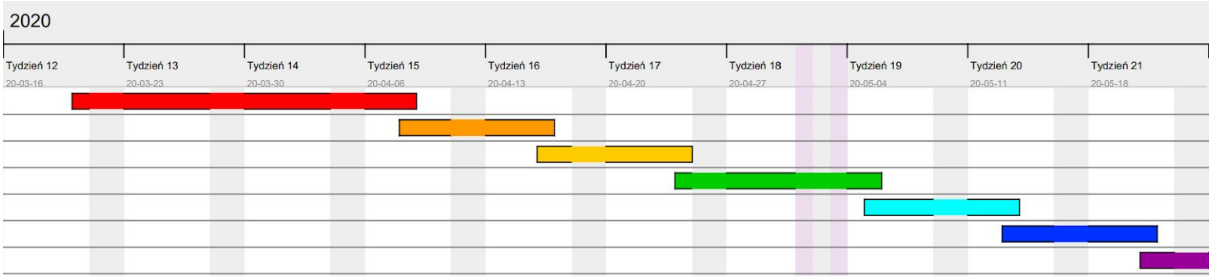
Poniżej przedstawiony jest harmonogram projektu w postaci wykresu Gantta, jak i diagramu osób.

Projekt Bazy Danych Do Aplikacji Android

Wykres Gantt'a

Nazwa zadania	Data roz...	Data zakończenia
Projekt bazy danych	20-03-20	20-04-08
Stworzenie bazy danych	20-04-08	20-04-16
Kontakt między aplikacją a bazą	20-04-16	20-04-24
Testy poprawności wprowadzania danych do bazy	20-04-24	20-05-05
Testy na bazie z resztą modułów	20-05-05	20-05-13
Dopaswanie bazy danych pod zmiany	20-05-13	20-05-21
Ostatnie zmiany i poprawki	20-05-21	20-05-29

Rys.2



Rys.3

11. Plan testów

Do przetestowania bazy danych, będziemy musieli sprawdzić czy istnieje możliwość dodania rekordów, modyfikacji, usunięcia za pomocą zapytań SQL. Wraz z możliwościami potrzebne będzie stworzenie zapytań, które będą zwracały wyniki możliwości dodawania oraz edycji tabel jak i rekordów. Testowana będzie również wielkość tabel jak i wielkość bazy danych odpowiednimi formami zapytań.

12. Plan bezpieczeństwa

Room Persistence Library pomaga stworzyć pamięć podręczną danych aplikacji na urządzeniu, na którym jest ona uruchomiona. Ta pamięć podręczna, która służy jako główne źródło zabezpieczeń w aplikacji, umożliwia użytkownikom przeglądanie spójnej kopii kluczowych informacji w aplikacji, niezależnie od tego, czy użytkownicy mają połączenie z Internetem. Oznacza to że użytkownik zawsze pracuje na kopii zapasowej/widoku bazy danych, a nie na samej bazie co zwiększa

bezpieczeństwo i minimalizuje ryzyko dodanie nieprawidłowych danych, które mogłyby uszkodzić bazę danych jak np. wymuszone SQL Injection.

13. Plan konserwacji oraz warunków licencji

Niniejszym udziela się bezpłatnie każdej osobie, która otrzymuje kopię tego oprogramowania i powiązanych plików dokumentacji („Oprogramowanie”), do czynienia z Oprogramowaniem bez ograniczeń, w tym między innymi prawa do używania, kopiowania, modyfikowania, łączenia , publikowania, rozpowszechniania, udzielania podlicencji i / lub sprzedawać kopie Oprogramowania oraz zezwalać na to osobom, którym Oprogramowanie zostało dostarczone, z zastrzeżeniem następujących warunków: Powyższa informacja o prawach autorskich i niniejsza informacja o pozwoleniu będą zawarte we wszystkich kopiach lub znacznych częściach Oprogramowania. OPROGRAMOWANIE JEST DOSTARCZANE „W STANIE, W JAKIM JEST”, BEZ ŻADNEJ GWARANCJI, WYRAŹNEJ LUB DOROZUMIANEJ, W TYM, ALE NIE OGRANICZONE DO GWARANCJI PRZYDATNOŚCI HANDLOWEJ, PRZYDATNOŚCI DO OKREŚLONEGO CELU I NARUSZENIA. W ŻADNYM WYPADKU AUTORZY LUB POSIADACZE PRAW AUTORSKICH NIE PONOSZĄ ODPOWIEDZIALNOŚCI ZA JAKIEKOLWIEK ROSZCZENIE, SZKODY LUB INNE ODPOWIEDZIALNOŚCI, NAWET W DZIAŁANIU UMOWY, TORTU LUB INNYCH INNYCH DZIAŁALNOŚCI, WYNIKAJĄCE Z, LUB ZWIĄZANE Z OPROGRAMOWANIEM LUB WYKORZYSTANIEM LUB INNYMI OKREŚLENIAMI OPROGRAMOWANIE.