Rok akademicki 2018/2019

Uniwersytet Mikołaja Kopernika w Toruniu

Wydział Matematyki i Informatyjki



PRACA DYPLOMOWA INŻYNIERSKA

Jakub Dziennik

Aplikacja do zarządzania

zdrowiem na system Android

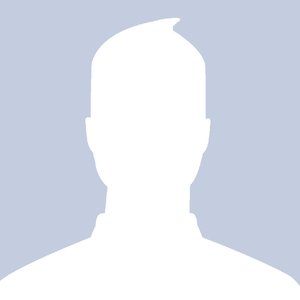
Opiekun pracy

Marta Burzańska

Ocena: ....................................................

................................................................

Podpis Przewodniczącego   
Komisji Egzaminu Dyplomowego



Imię i nazwisko: Jakub Dziennik

Kierunek: Informatyka

Data urodzenia: 1996.04.14

Data rozpoczęcia studiów: 2015.10.01

Życiorys

Urodziłem się 14 kwietnia 1996 roku w Bydgoszczy. Od 2012 roku uczęszczałem do I Liceum Ogólnokształcącego im. Bolesława Chrobrego o profilu mat-fiz-inf w Grudziądzu. W 2015 roku ukończyłem szkołę średnią. Studia na wydziale Matematyki i Informatyki rozpocząłem w październiku 2015r., wybierając kierunek Informatyka.

. . . . . . . . . . . . . . . . . .

Podpis studenta

EGZAMIN DYPLOMOWY

Złożył egzamin dyplomowy w dniu. . . . . . . . . . . . . . . . . . . . . . . . .2019r. z wynikiem . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Ogólny wynik studiów: . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Dodatkowe wnioski i uwagi Komisji: . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

**Spis treści**

1. **Wstęp (3 strony)**
   1. Omówienie koncepcji (1 strona)
   2. Istniejące rozwiązania na rynku:

MediSafe, MyTherapy, Moje pigułki (1-2 strony)

1. **Narzędzia (5 stron)**
   1. Java (1 strona)
   2. Android (1 strona)
   3. Android Studio (1 strona)
   4. XML (0,5 strony)
   5. SQLite (0,5 strony)
   6. GitHub (0,5 strony)

**3. Implementacja (15 stron)**

4.1. Menu

4.2. Dodawanie alarmu

4.3. Rezultaty implementacji

**4. Testy (2 strony)**

**5. Podsumowanie (1 strona)**

**6. Bibliografia (1 strona)**

1. **Wstęp**

Rozwój technologii w dzisiejszych czasach biegnie w zaskakująco szybkim tempie. Telefon, który kiedyś służył tylko do kontaktu z ludźmi, obecnie posiada praktycznie te same funkcje co komputer. Nie ma znczenia czy potrzebujemy sprawdzić pogodę, czy dokonać wnikliwej analizy kursu walut. Większość z tych czynności możemy wykonać w ciągu kilku sekund włączając odpowiedni program lub stronę internetową. Ciężko spotkać osobę na ulicy, która nie posiada w kieszeni smartfona. W związku z bardzo dużą popularnością smartfonów rośnie również popyt na aplikacje mobilne.

Dlatego też, w niniejszej pracy omówiono proces tworzenia aplkacji do zarządzania zdrowiem. Program został zaimplementowany w języku Java na system Android, który został stworzony kilka lat temu przez markę Google. Jest to jeden z najbardziej popularnych platform na nowoczesny telefon komórkowy. Aplikacja przeznaczona jest między innymi dla osób, które regularnie przyjmują leki. Program posiada kilka funkcji, są to między innymi: przypominanie o wzięciu tabletki, wizycie u lekarza, dodawanie wyników pomiarów badań. Osoba korzystająca z aplikacji jest w stanie stworzyć kilka profili, dzięki czemu może zarządzać zdrowiem swoich najbliższych osób.

W pierwszym rozdziale zawarto omówienie koncepcji projektu oraz cel i zakres pracy. Rozdział drugi został poświęcony na analizę istniejących rozwiązań na rynku. Tematem trzeciego rozdziału są użyte technologie w projekcie: Java, Android Studio, Android, SQLite. Rodział czwarty został poświęcony na minimalne wymagania, implementację systemu oraz wyglądu aplikacji. Zawarto w nim między innymi zarządzanie bazą danych, dodawanie alarmu oraz kilku profili użytkowników. W rozdziale piątym przedstawiono testy aplikacji wraz z oceną wyglądu oraz funkcjonalności przez różne osoby korzystające z programu.

1. **1. Omówienie koncepcji**

Tematem mojej pracy jest aplikacja mobilna do zarządzania zdrowiem. W poniższym rozdziale chciałbym omówić koncepcję mojego projektu. Oprócz tego przedstawię wymagania systemu konieczne do użytkowania aplikacji. Docelową platformą jest smartfon z systemem Android w wersji co najmniej 7.0. System Android został wybrany ze względu na bardzo dużą popularność. Według statystyk w 2017 roku użytkownicy dokonali 19 miliardów pobrań aplikacji ze sklepu Google Play. Dzięki takiemu środowisku jesteśmy w stanie dotrzeć do bardzo dużej ilości potencjalnych użytkowników naszej aplikacji.

Główną funkcją „Weź pigułkę” jest przypomnienie o wzięciu tabletki. Program o podanej godzinie informuje użytkownika o konieczności wzięcia tabletki wraz z dwoma przyciskami: „Wziąłem” oraz „Zapomniałem”. Każde kliknięcie przycisku zostaje wysłane do systemu, który zapisuje statystyki użytkownika. Użytkownik ma możliwość wygenerowania raportu w PDF, który zawiera listę wziętych i niewziętych tabletek. Dzięki temu jesteśmy w stanie sprawdzić całą historię naszego leczenia. Osoba korzystająca z aplikacji może szybko sprawdzić postęp wziętych tabletek w zakresie procentowym odpowiednio od 0 do 100%. Podczas wysyłania przypomnienia aplikacja monitoruje ilość tabletek w apteczce. Jeśli zbliża się okres wyczerpania naszych tabletek otrzymujemy stosowne powiadomienie o uzupełnieniu zapasu. Użytkownik w każdej chwili ma możliwość sprawdzenia ilości tabletek danego leku. Oprócz przypominania o wzięciu tabletki aplikacja może przypomnieć o nadchodzącej wizycie u lekarza. Użytkownik otrzyma informację dzień przed wizytą oraz bezpośrednio 2 godziny przed. Dodatkowo można stworzyć własną listę lekarzy z danymi kontaktowymi. Kliknięcie na danego lekarza pozwala na wykonanie połączenia telefonicznego. Osoba korzystająca z aplikacji ma możliwość stworzenia własnych notatek np. listę zakupów z apteki lub zalecenia otrzymane od lekarza. Oprócz tego użytkownik może wpisywać do aplikacji wyniki dokonanych badań np. ciśnienia lub krwi. Jest to bardzo ważna funkcjonalność, ponieważ większość osób przyjmujących tabletki prowadzi codzienne badania w domu. Jeśli chcemy zarządzać zdrowiem kilku lub kilkunastu osób również nie ma z tym problemu. Istnieje możliwość dodania kilku profili i szybkiego przełączania między nimi. Dzięki temu użytkownik jest w stanie monitorować zdrowie każdego członka rodziny. W celu usunięcia przypomnienia lub profilu wystarczy przesunąć daną rubrykę w prawo i potwierdzić chęć usuwania. Aplikacja przewiduje błędy użytkownika np. informuje o niewypełnieniu danego pola lub o chęci dodania przypomnienia ze starą datą.

Program posiada prostą i przejrzystą szatę graficzną, która zakłada korzystanie z aplikacji przez osoby starsze. Dodatkowym atutem jest całkowity brak reklam, co jest rzadko spotykane w darmowych programach.

* 1. **Istniejące rozwiązania na rynku**

**2. Narzędzia**

W niniejszym rozdziale zostanie poruszona kwesta technologii wykorzystanych podczas tworzenia aplikacji. Większość z tych narzędzi jest kluczowa dla prawidłowego działania całego systemu.

**2.1. Java**

Java to jeden z najpopularniejszych języków programowania, który został oparty na klasach. Stworzony został w 1995 roku w firmie Sun Microsystems pod przewodnictwem Jamesa Goslinga. Java wywodzi się między innymi z języka C++, z którego zaczerpnięto część składni i słów kluczowych. Pomysł wirtualnej maszyny oraz koncepcja zarządzania pamięcią został przejęty z języka Smalltalk. Twórcy scharakteryzowali kilkanaście koncepcji języka Java. Kluczowe i najważniejsze z nich to:

* **Obiektowość**

Język Java w przeciwieństwie do języka C++ kładzie bardzo duży nacisk na programowanie obiektowe. Oznacza to, że podczas pisania kodu źródłowego pracujemy na obiektach i klasach. „Klasy – definicje formatu danych oraz dostępnych procedur dla danego typu lub klasy obiektu, mogą same także zawierać dane oraz procedury (znane jako metody klas), na przykład, klasy zawierają dane oraz metody. Obiekty – instancje klas.”. Dzięki wykorzystaniu obiektowości mamy możliwość łatwej modyfikacji kodu oraz dalszej rozbudowy. Opróćz tego przejrzystość naszego kodu znacznie wzrasta.

* **Dziedziczenie**

W języku Java mamy do czynienia z dziedziczeniem, dzięki czemu jedna klasa może dziedziczyć zmienne lub metody po drugiej klasie. Klasą pochodną nazywamy klasę dziedziczącą, a klasą bazową nazywamy klasę, po której dziedziczymy. Istnieją 3 typy dziedziczenia: publiczne (public), chronione (protected) oraz prywatne (private). Wykorzystując dany typ dziedziczenia mamy możliwość ograniczenia widoczności elementów z klasy. W języku Java odstąpiono od dziedziczenia wielokrotnego, dzięki czemu wyeliminowano możliwość wystąpienia konflików.

* **Niezależność od architektury**

Kod źródłowy programów napisanych w języku Java kompiluje się do kodu pośredniego. Kod pośredni zostaje wykonywany przez tak zwaną wirtualną maszynę Javy, która tłumaczy kod na kod dostosowany do określonego procesora i systemu operacyjnego. Dzięki takiemu rozwiązaniu nasz program nie jest ograniczony od jednostki.

* **Sieciowość**

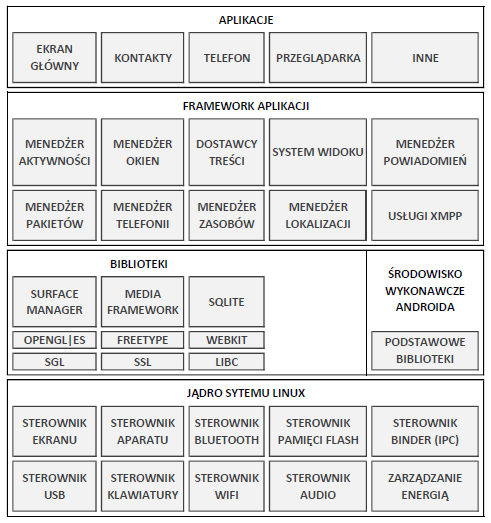
Java w swoich zasobach posiada bibliotekę, dzięki której mamy możliwość korzystania z protokołów sieciowych, między innymi HTTP oraz FTP. Dla języka nie ma różnicy pomiędzy korzystaniem z pliku na dysku i na serwerze sieciowym.

* **Niezawodność i bezpieczeństwo**

Głównym założeniem języka od początku swojego istnienia było zastąpienie języka C++. Twórcy mieli na celu stworzenie języka bezpiecznego oraz niezawodnego. W związku z tym w Javie mamy do czynienia z wyjątkami. Wyjątki pozwalają zareagować kiedy istnieje ryzyko niewykonania danego zadania np. operacje na elemencie poza tablicą.

**2.2. Android**

Android jest jednym z najbardziej popularnych systemów operacyjnych na urządzenia mobilne takie jak telefony komórkowe, smartfony, tablety oraz netbooki. Według statystyk na świecie używanych jest obecnie miliard urządzeń z powyższym systemem. Android posiada otwarty kod źródłowy oraz oparty jest na jądrze Linux. Pierwsze wydanie systemu nastąpiło w 2008 roku. Obecnie wspierany jest przez firmę Google. Dzięki takiej kompleksowej platformie mamy idealne podłoże do tworzenia rozbudowanych aplikacji. Do tego celu używamy języka Java oraz XML, które zostały omówione w dalszych podpunktach. Spójrzmy na schemat budowy tego systemu (Rysunek X).



U podnóża systemu znajduje się jądro systemu Linux. Odpowiada ono między innymi za obsługę sterowników oraz usług takich jak zarządzanie energią oraz zabezpieczenia.

Powyżej warstwy jądra mamy do czynienia z warstwą zestawu bibliotek C/C++. Występują tam między biblioteki do zarządzania bazą danych, grafiką oraz silnik przeglądarki internetowej.

Na tym samym poziomie zostało wyodrębnione środowisko wykonawcze Android. Wyposażone jest ono w podstawowe biblioteki, które zapewniają większość funkcjonalności języka Java.

W kolejnej warstwie mamy do czynienia z frameworkiem aplikacji, która daje nam dostęp do podstawowych API używanych podzcas tworzenia oprogramowania mobilnego. Dzięki temu możemy odczytać lokalizację telefonu lub pokazać powiadomienie z naszej aplikacji.

Na ostatnią warstwę składa się warstwa aplikacji w które wyposażony jest system Android. W skład wchodzą aplikacje domyślne, takie jak aparat, kontakty, ale również te zainstalowane ze sklepu Google Play lub innych źródeł.

Podsumowując system Android posiada dosyć złożoną budowę, jednak ze względu na ilość udostępnionych narzędzi programistycznych oraz bibliotek stanowi on doskonałą platforme do tworzenia aplikacji mobilnych.

**2.3. Android Studio**

Android Studio jest środowiskiem programistycznym opartym na IntelliJ IDEA, które służy do tworzenia aplikacji na Androida. Zaprezentowano je na konferencji Google w 2013 roku. Od początku istnienia cieszyło się ogromnym zainteresowaniem ze strony początkujących oraz zaawansowanych developerów. Powyższe środowisko charakteryzuje się rozbudowanym interfejsem oraz bogatą funkcjonalnością. Kompilacja oparta jest na systemie Gradle, który przetwarza kod źródłowy projektu, zasoby i łączy w plik APK gotowy do instalacji na naszym telefonie. Android Studio posiada również dostęp do gotowych szablonów oraz bibliotek wspierających tworzenie aplikacji. Układ programu tworzony jest w języku XML, który został omówiony w dalszym podpunkcie. Dodatkowo mamy możliwość testowania wyglądu w różnych rozdzielczościach bez konieczności kompilowania naszego projektu. Dzięki takiemu rozbudowanemu środowisku jesteśmy w stanie stworzyć wymarzony projekt nawet będać zwykłym laikiem.

**2.4. XML**

XML (Extensible Markup Language) jest uniwersalnym językiem znaczników, który służy do reprezentowania przeróżnych danych w strukturalizowany sposób. XML został opracowany w 1996 roku przez grupę kierowaną przez Jona Bosaka. Popularność języka zawdzięcza łatwej konwersji dokumentów między przeróżnymi systemami. Dzięki temu XML nie jest uzależniony od używanej platformy. W Android Studio pliki XML służą programiście do zaprojektowania układu poszczególnych elementów ekranu dla każdej aktywności w aplikacji. Układ ustalany jest poprzez klasę Activity za pomocą metody setContentView (View) wewnątrz metody OnCreate (Bundle). Podczas projektowania układu w edytorze tekstu, Android Studio umożliwia podgląd, dzięki czemu możemy zobaczyć rezultat modyfikowania kodu XML w czasie rzeczywistym. Podczas projektowania układu aplikacji w trybie projektowania Android Studio automatycznie zamieni interfejs graficzny na kod źródłowy.

**2.5. GitHub**

GitHub jest darmowym programem, który śledzi zmiany w plikach tak, że określone wersje mogą zostać później przywrócone. GitHub został uruchomiony w 2008 roku i z roku na rok jego popularność rosła. Obecnie korzysta z niego ponad 31 milionów użytkowników. Zainteresowanie tym rozwiązaniem wynika między innymi ze względu na łatwość użytkowania. W celu rozpoczęcia naszej przygody konieczne jest stworzenie konta oraz zdalnego repozytorium. Zdalne repozytorium to wersja projektu, która jest przechowywana na serwerze. Istnieje możliwość posiadania kilku repozytoriów w zależności od preferencji. GitHub jest idealny przy współpracy kilku(nastu) osób nad jednym projektem. Aby pobrać najnowszą wersję naszej aplikacji wystarczy użyć komendy **git pull**, która pobiera aktualne dane z serwera i scala z naszym kodem roboczym. W celu wypchnięcia swoich zmian do serwera z pomocą przychodzi nam komenda **git push**. Polecenie zadziała wyłącznie wtedy, gdy posiadamy najnowszą wersję na naszym komputerze. Istnieje możliwość podglądu wszystkich wysłanych wersji na serwer, dzięki czemu możemy sprawdzić postępy w implementacji naszej aplikacji. Oprócz tego mamy możność do przwyrócenia wcześniejszej wersji.

**2.6. SQLite**

SQLite jest bardzo popularną i darmową biblioteką programistyczną, która została stworzona w 2000 roku przez Richarda Hippa w języku C. Biblioteka implementuje silnik SQL, który umożliwia korzystanie z baz danych bez uruchamiania osobnego procesu. SQLite czyta i zapisuje dane jako pliki binarne. Bezpieczeństwo tych plików oparte jest na zabezpieczeniach, które oferuje używany przez nas system plików. Dzięki temu zawartość bazy danych może być łatwo przenoszona między różnymi platformami. SQLite obsługuje między innymi: widoki, transakcje, klucze obce, wyzwalacze oraz zapytania zagnieżdżone. Wydajność w porównaniu z innymi serwerami baz danych jest bardzo wysoka. Na SQLite oparte jest wiele popularnych aplikacji na system Windows, Android, Windows Phone oraz iOS. Według zgromadzonych statystyk ponad bilion danych SQLite jest aktywnych na świecie.

**6. Bibliografia**

<https://pl.wikipedia.org/wiki/SQLite>

<https://www.dobreprogramy.pl/SQLite,Program,Windows,86416.html>

<https://sqlitestudio.pl/tcltut/tutorial/data/addons/db/sqlite.html>

<https://pl.wikipedia.org/wiki/GitHub>

<https://git-scm.com/book/pl/v1/Podstawy-Gita-Praca-ze-zdalnym-repozytorium>

<https://pl.wikipedia.org/wiki/XML>

<http://www.samouczekprogramisty.pl/xml-dla-poczatkujacych/>

<https://pl.wikipedia.org/wiki/Java>

<https://www.instalki.pl/programy/download/Windows/srodowiska_programistyczne/Android_Studio.html>