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

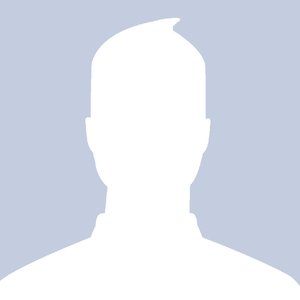
Opiekun pracy

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. Cel i założenia projektu (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. Specyfikacja wymagań systemu (1 strona)**

3.1. Wymagania funkcjonalne

3.2. Wymagania niefunkcjonalne

**4. Implementacja (15 stron)**

4.1. Menu

4.2. Dodawanie alarmu

4.3. Rezultaty implementacji

**5. Testy (2 strony)**

**6. Podsumowanie (1 strona)**

**7. 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 rynku giełd. 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 spodni lub w torebce 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. Cel i założenia projektu**

Celem pracy jest zaprojektowanie oraz zaimplementowanie aplikacji, która pozwoli użytkownikowi w prosty sposób zarządzać zdrowiem. Docelową platformą jest smartfon z systemem Android. Funkcje aplikacji prezentują sie następująco: przypomnienie o wzięciu tabletki wraz z ilością tabletek w apteczce, wyniki dokonanych badań (krwi, ciśnienia itp), lista lekarzy, leków, własne notatki oraz możliwość dodania kilku profili. Program o podanej godzinie informuje użytkownika o konieczności wzięcia tabletki wraz z dwoma przyciskami: „Wziąłem” oraz „Zapomniałem”. Program zapisuje status dla danego przypomnienia oraz monitoruje stan tabeletek. W przypadku zbliżającego sie wyczerpania tabletek w apteczce osoba korzystająca z aplikacji otrzyma stosowne powiadomienie na ekranie. Dodatkową funkcjonalnością będzie możliwość wygenerowania raportu w PDF, który zawiera listę wziętych i niewziętych tabletek. Oprócz tego użytkownik szybko będzie mógł sprawdzić stan postępu wziętych tabletek w zakresie procentowym odpowiednio od 0 do 100%. Program powinien posiadać przejrzystą szatę graficzną zakładając korzystanie z aplikacji przez osoby starsze. Dodatkowym atutem będzie 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**

**2.2. Android**

**2.3. Android Studio**

**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.