1. Wstęp.

Praca inżynierska podzielona jest na dwie części. Składa się z części praktycznej czyli, z aplikacji na telefon Android i IOS oraz z części teoretycznej podzielonej na rozdziały. W tym rozdziale przedstawię jak pojawiła się koncepcja na powstanie pracy i dlaczego wybrałem ten temat pracy.

Temat pracy jest umotywowany zainteresowaniem, autorem nowymi technologiami w szczególności smartfonami oraz oprogramowaniem działających w nich. Rynek smartfonów i innych smart-urządzeń się coraz bardziej rozrasta. Większość ludzi do przeglądania zasobów internetu i komunikacji wykorzystuje smartfony. Dlatego tworzenie aplikacji w obecnej sytuacji jest bardzo opłacalne.

Drugą motywacją autora są ciągle dojazdy czy to do pracy czy na uczelnię, a co za tym idzie możliwość spóźnienia na autobus, aplikacja, która działa w tle i przypomina o tym, że powinniśmy wyjść na autobus, czy pociąg, żeby się nie spóźnić.

1. Cel oraz Koncepcja.

Celem pracy jest przygotowanie aplikacji, która dotrze do większości użytkowników systemów mobilnych (Android + IOS 99,8%).

Powinna być intuicyjna w taki sposób, aby każdy mimo braku znajomości języka polskiego czy angielskiego mógł ją obsługiwać. Aplikacja powinna być przygotowana w technologii, która pozwala na jednoczesne tworzenie aplikacji na system Android i IOS.

Aplikacja powinna być przejrzysta i przyjemna w obsłudze dla użytkownika. Interfejs powinien być tak zaprojektowany, że niezależnie od platformy czy wielkości wyświetlacza, będzie wygodny dla użytkownika.

Powiadomienia przypominające o wyjściu na autobus, pociąg itd powinny się pojawiać niezależnie czy aplikacja, jest uruchomiona czy wyłączona. Nie powinno to też mieć znaczenia połączenie do Internetu.

Wpisane przez użytkownika przypomnienia powinny być synchronizowane w bazie danych, do której dostęp będzie miał tylko użytkownik.

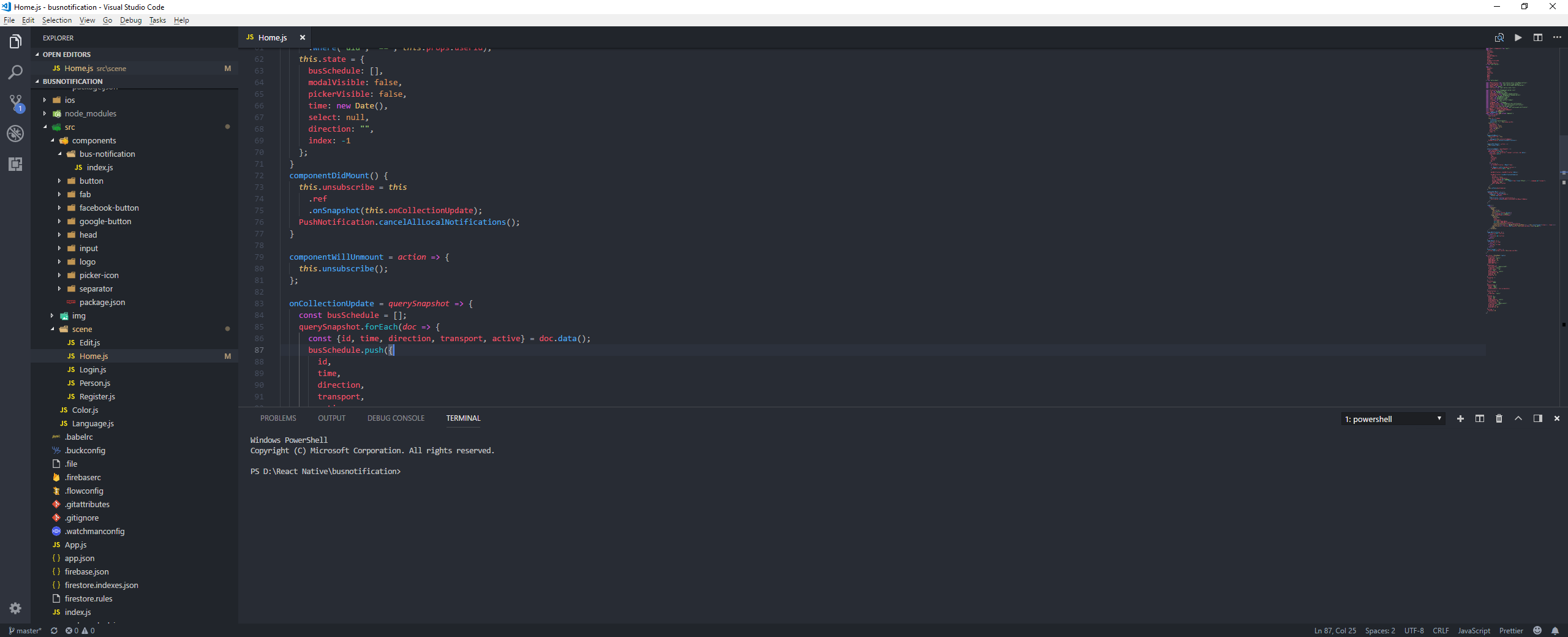
1. Środowisko produkcyjne.

Do stworzenie części praktycznej tej pracy autor użył następujących narzędzi:

Visual Studio Code, czyli darmowy edytor kodu, działający na systemach Windows, Linux i MacOS.

Tworząc pracę autor wybrał ten program ze względu na:

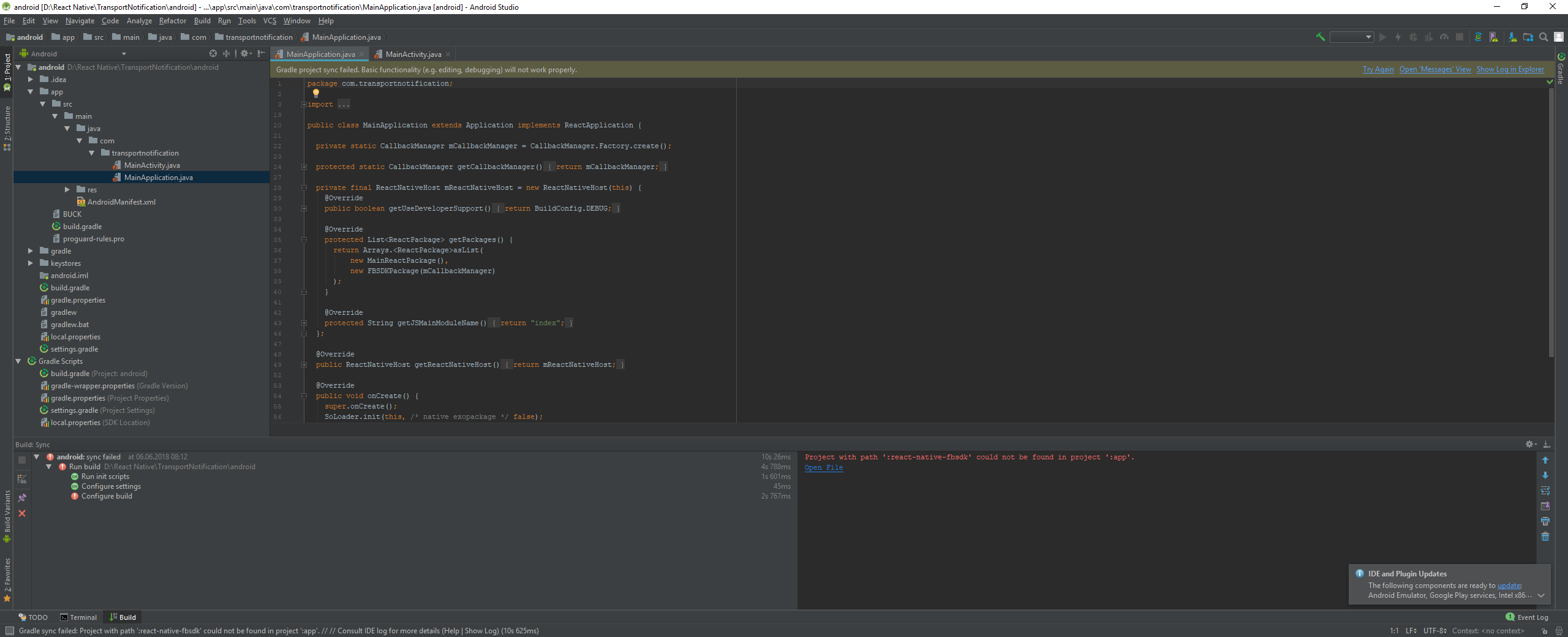
* Szybkość działania
* Dużą ilość dodatków ułatwiających pisanie
* Przejrzystość interfejsu



Rysunek przedstawiający program Visual Studio Code.

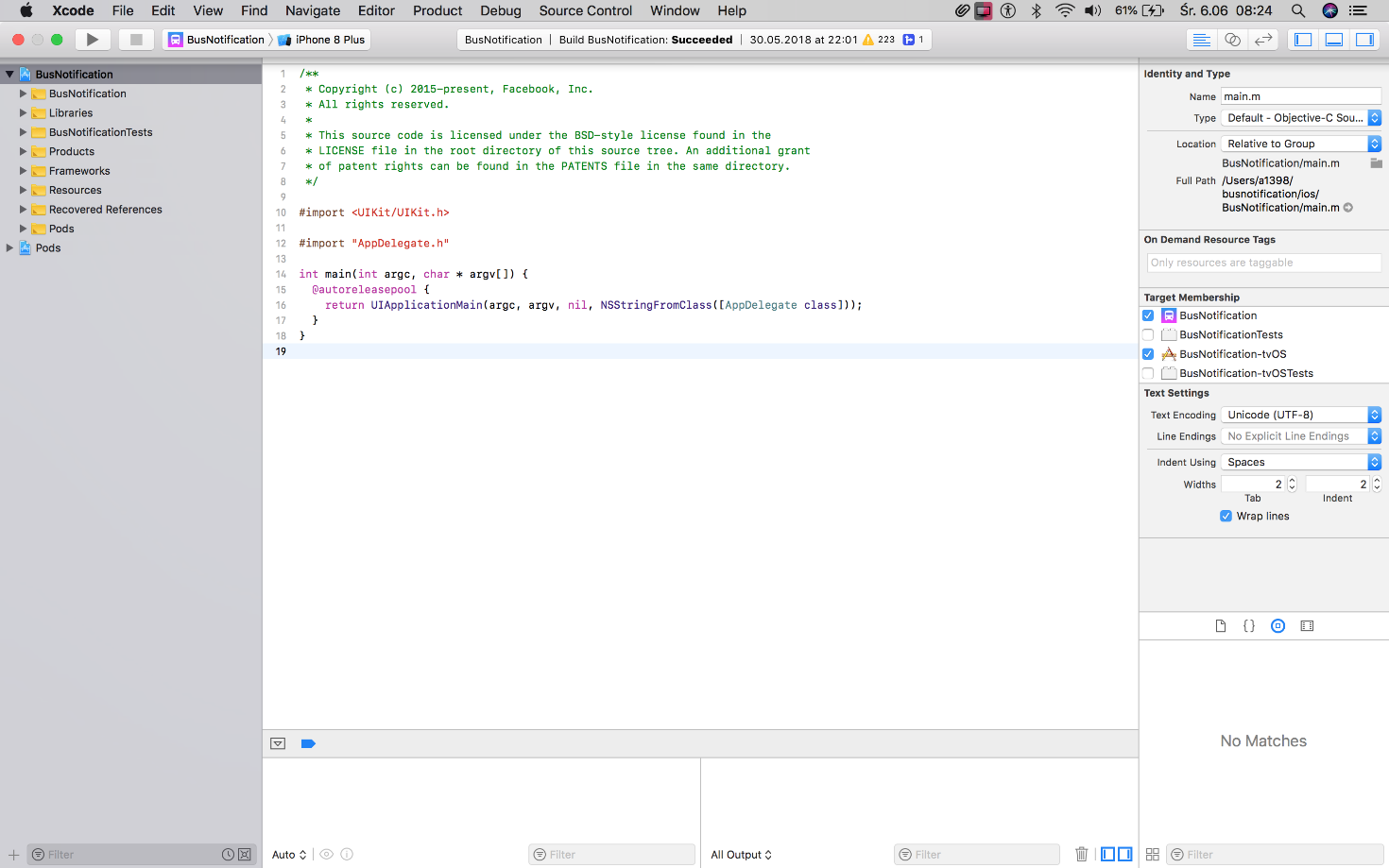
React Native Tools, darmowy dodatek do VSC pozwalający na pisanie aplikacji w języku React Native w VSC, wybór tego narzędzie był podyktowany wyborem edytora.

Android Studio, darmowy edytor kodu platformy Android, pozwalający na tworzenie aplikacji w języku Java oraz Kotlin. Autor użył tego programu do łączenie komponentów natywnych w języku Java z React Nativem.



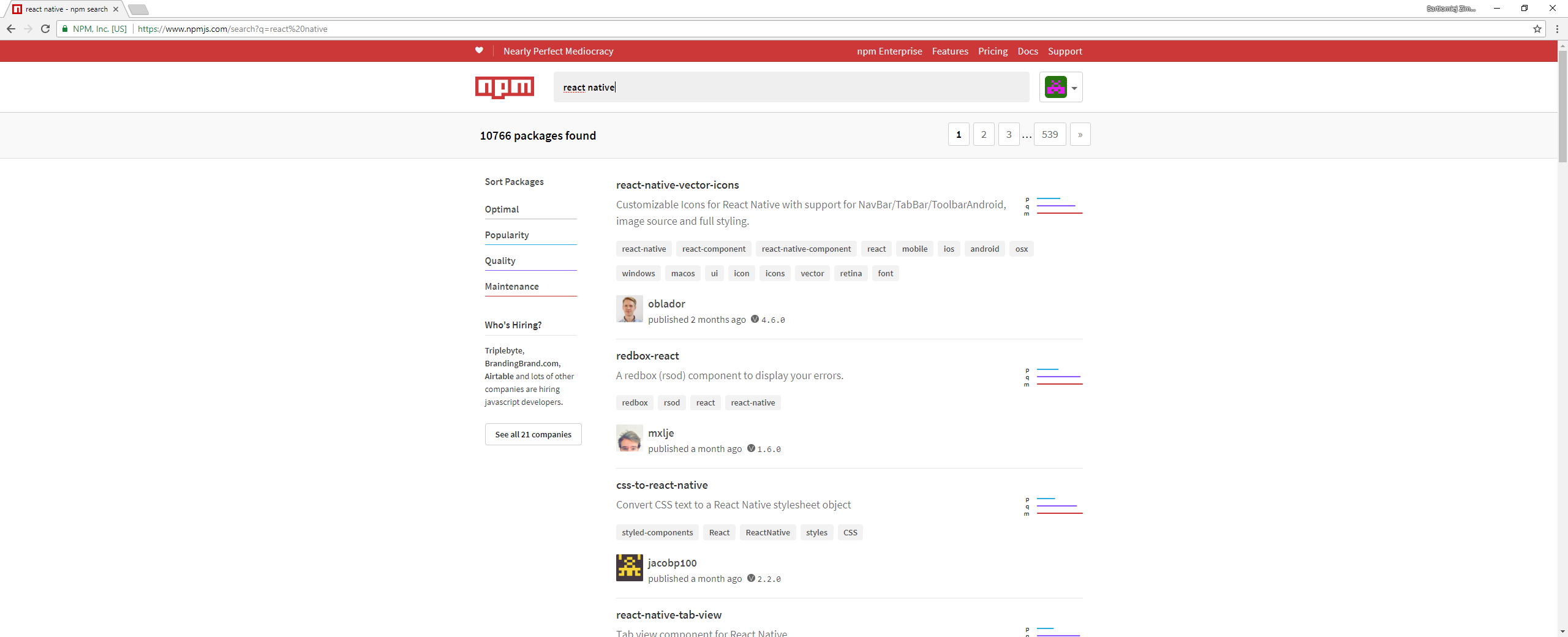
Rysunek przedstawiający program Android Studio

XCode, darmowy edytor kodu platformy IOS, pozwalający na tworzenie aplikacji w języku Swift oraz Objective C. Autor użył tego programu do łączenie komponentów natywnych w języku Objective C z React Nativem.



Rysunek przedstawiający program XCode.

Menedżer pakietów Npm, jest to całe środowisko składające się z strony npmjs.com wyświetlające listę pakietów, baza pakietów dla języka Javascript i frameworków opierających się o język JavaScript oraz klient na systemy z Windows, Linux i MacOS, pozwalający na zarządzanie pakietami. Npm ułatwia tworzenie kodu, ponieważ daje łatwy dostęp do ogromnej liczby darmowych pakietów, z których można skorzystać, przy tworzeniu aplikacji.

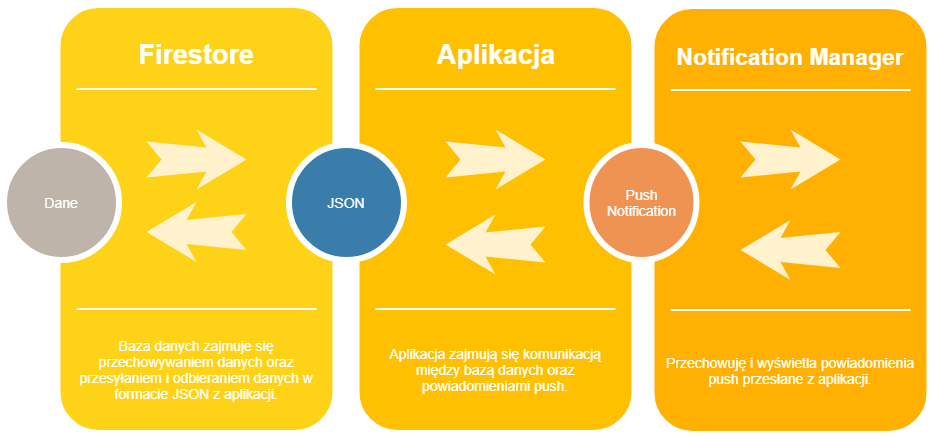


Rysunek przedstawiający katalog pakietów na stronie npmjs.com.

1. Project Aplikacji

##### Architektura

##### Architektura aplikacji składa się z 3 części pakietu Firebase, aplikacji napisanej w React Native, Notification Manager. Dane są przechowywane w bazie danych Firestore, należących do pakietu Firebase. Aplikacja komunikuje się z bazą za pomocą formatu JSON z wykorzystaniem biblioteki react-native-firebase, Aplikacja pobrane komunikuje się między Firestorem, a Notification Managerem, gdzie znajdują się powiadomienia.



Rysunek przedstawia w jaki sposób przekazywane są dane w aplikacji BusNotification.

Widoki

Aplikacja będzie się składać z co najmniej 5 widoków:

Widok logowania, za pomocą, którego można się będzie zalogować do aplikacji.

Widok rejestracji za pomocą, którego można się będzie zarejestrować do aplikacji.

Widok listy powiadomień, gdzie będzie można zobaczyć dostępne powiadomienia i je włączyć.

Widok dodania i edycji powiadomienia.

Widok wylogowania i zmiany hasła.

Powiadomienia

Powiadomienia powinny działać w taki sposób, aby użytkownik po uruchomieniu dostawał powiadomienia codziennie o wskazanej przez siebie godzinie, nawet jeśli będzie miał wyłączoną aplikacji i będzie bez dostępu do Internetu.

1. Założenia aplikacji.

Główne założenia aplikacji BusNotification:

* Prostota użytkowania
* Szybkość działania
* Wieloplatformowość
* Możliwość obsługi w kilku językach
* Łatwy sposób tworzenia konta w aplikacji
* Nowoczesny i uniwersalny interfejs aplikacji

1. Funkcje Programu

##### Założenia funkcjonalne:

##### Logowanie za pomocą mailu, konta Facebook oraz konta Gmail za pomocą Firebase Auth

##### Rejestracja za pomocą mailu i hasła

##### Synchronizacja danych między bazą danych a aplikacją

##### Używanie lokalnych powiadomień push do przypominania o transporcie

##### Możliwość zmiany częstotliwości powiadomień

##### Kilka języków obsługiwania angielski, polski

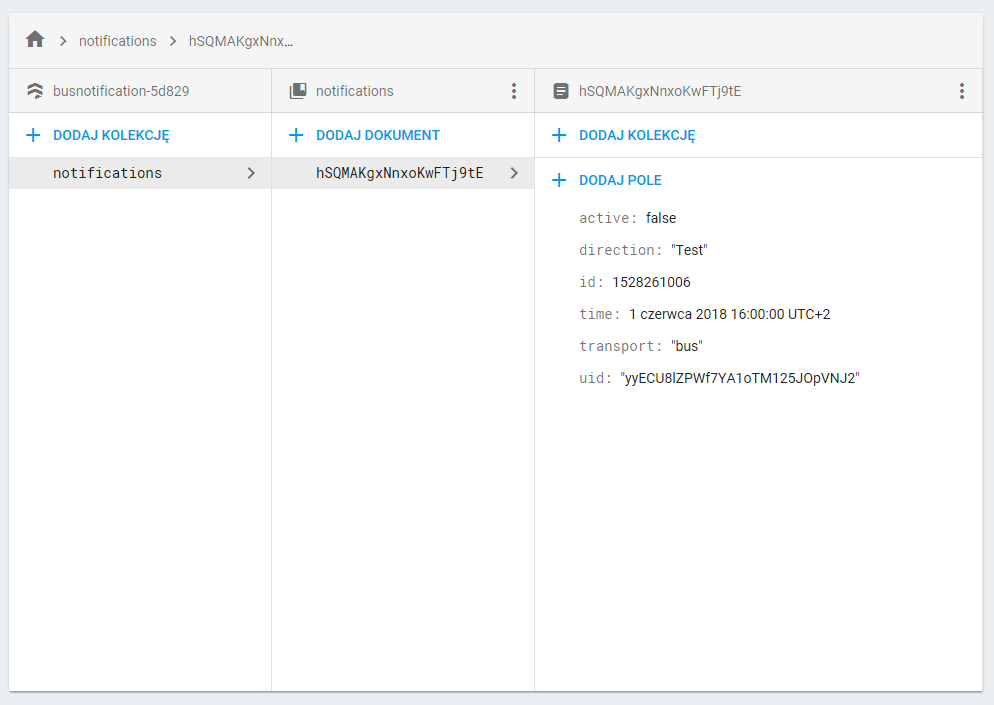
##### Aplikacja działająca na Android i IOS

Wymagania niefunkcjonalne:

1. Aplikacja będzie działać w czasie rzeczywistym, tzn. zmiany dokonane przez użytkownika, będą od razu zapisywane w bazie danych.
2. Aplikacja nie będzie skomplikowana w obsłudze.
3. Aplikacja będzie mieć nowoczesny wygląd przyciągający użytkownika.
4. Wygląd interfejsu będzie uniwersalny bez względu na platformę.
5. Aplikacja będzie szybka w działaniu.
6. Aplikacja będzie łatwa w rozbudowie.
7. Aplikacja będzie pozwalać na łatwe wykrywanie i rozwiązywanie błędów.
8. Project Aplikacji.

Autor pracy wybrał zestaw narzędzi Firebase, udostępniający funkcje takie jak logowanie i rejestracja użytkowników, analiza i wykrywanie błędów w aplikacji oraz dwie bazy danych Realtime Database oraz Cloud Firestore.

Autor pracy jako bazę danych przechowującą pliku użytkowników wybrał Cloud Firestore. Jest to baza danych oparta o NoSQL. Jest bazą, która opera się do dokumentach i kolekcjach. Dokument to jeden zestaw przechowanych danych. Dokumenty grupuje się w kolekcje.



Baza składa się z jednej tabeli notifications (Powiadomienia). Znajdują się w niej active (Aktywny) oznaczenie czy powiadomienie jest włączone, direction (Kierunek podróży), id: (identyfikator powiadomienia) generowany na postawie czasu w notacji unixowej, time (godzina, w której zostanie wyświetlone powiadomienie), transport (środek transportu), uid (identyfikator użytkownika). Poza tym każdy dokument ma są uid pozwalający na wyróżnienie go na liście z dokumentami.

1. Opis rozdziałów / Struktura pracy
2. Koncepcja aplikacji oraz narzędzia.
   1. Środowisko produkcyjne.
3. Analiza wymagań oraz projekt aplikacji.
4. Założenia aplikacji.
5. Funkcje programu.
   1. Ogólne wymagania funkcjonalne.
   2. Ogólne wymagania niefunkcjonalne.
6. Projekt aplikacji:
   1. Baza danych.
   2. Opis tabel.
7. Javascript
8. React Native
9. CSS
10. Npm
11. Jsx
12. State i Props
13. Różnice między IOS i Android
14. Komponenty natywne
15. Komponenty React Native
16. Tworzenie Komponentów

Npm jest managerem pakietów javascript. Znajduję się w nim duża ilość darmowych pakietów, które można użyć w swoich programach. Pakiety są tworzone przez użytkowników, mogą być publiczne do użytku dla wszystkich albo prywatne dla użytku osobistego. Pakiet jest zbiór funkcji albo komponentów pozwalający ułatwić pisanie oprogramowania. Darmowe konto na npmjs.con pozwala na tworzenie tylko publicznych pakietów. Wbudowana wyszukiwarka pozwala na łatwe wyszukiwanie najlepiej ocenianych pakietów.Npm używałem w przy tworzeniu aplikacji do pobraniu gotowych pakietów, które ułatwiły i przyśpieszyły tworzenie aplikacji.

Podstawowe komendy

npm install

npm install nazwa pakietu –save

npm install npm –global

plik package.json

W tym pliku znajdują się wszystkie zainstalowane za pomocą klienta pakiety. Można w nim łatwo kontrolować liczbę użytych pakietów w aplikacji oraz aktualizację wersji pakietów.

plik package-lock.json

W tym pliku znajdują się wszystkie zależności między pakietami pobranymi za pomocą npm. Pozwala przyśpieszyć instalację wszystkich pakietów, dzięki temu że nie są instalowane powtarzające się pakiety. Dzięki niemu można odtworzyć zawartość node-modules i kontrolować zmiany w jego zawartości.

Zasada działania serwisu

Serwis npmjs.con składa się z strony z bazy z pakietami do pobrania oraz klienta npm. Klient służy do poblikowsnia oraz pobierania pakietów z bazy. Za pomocą strony mamy dostęp do bazy z pakietami.

“Recykling kodu” Reusabe-code

Serwis wpisuje się w modułowy sposób tworzenia aplikacji. Tworząc aplikację piszemy, komponenty czy funkcję które można powtórnie wykorzystać w innym programie. Npm pozwala w łatwy sposób udostępniać swoje oprogramowanie. Organizacja serwisu pozwala na stworzenie własnego konta, stworzenie organizacji w której będziemy udostępniać swoje komponenty oraz prywatne publikowanie pakietów, są wtedy dostępne tylko dla twórcy, bądź organizacji w której zostały udostępnione. Niestety ta za tą usługę trzeba zapłacić.

Javascript jest to skryptowy język programowania, stworzony przez firmę Netscape. Twórcą języka JavaScript jest Brendan Eich. Został stworzony z myślą o tworzeniu stron internetowych, ale znajduje teraz zastosowanie podczas tworzenia aplikacji mobilnych czy aplikacji na komputery z systemem windows, mac os, linux.