1. Motywacja

Temat pracy jest umotywowany zainteresowaniem Autora nowymi technologiami w szczególności smartfonami oraz oprogramowaniem działających na takich platformach. 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. Z aplikacją mobilną łatwiej się dotrzeć do potencjalnych użytkowników, ponieważ w aplikację na daną platformę znajdują się w sklepie, z którego można pobrać lub zakupić daną aplikację.

Drugą motywacją autora są spóźnienia na autobus, spowodowane, zbyt późnym wyjściem z domu lub roztargnieniem. Autor w tym celu wymyślił aplikację, która będzie miała przypominać ludziom, że już powinni wyjść na autobus, pociąg itd.

1. Cel.

Celem pracy jest przygotowanie aplikacji w frameworku React Native z wykorzystaniem platformy Google Firebase.

Praca ma przedstawić nowy sposób pisania hybrydowych aplikacji mobilnych pisanych w języku Javascript oraz wykorzystania platformy Firebase do autoryzacji użytkowników, bazy danych i testowania aplikacji.

React Native jest szkielet do tworzenia aplikacji mobilnych przez firmę Facebook w oparciu o framework webowy React.

React Native posiada dodatek, który umożliwia łatwą pracę z platformą Firebase od firmy Google.

Google Firebase jest platformą, która może zastąpić bazę danych, autoryzację, wykrywanie błędu, czy przechowywanie plików. Podstawowe funkcjonalności z wersji ograniczonej są dostępne za darmo, za dodatkowy transfer, czy zwiększenie odpowiedni na zapytania do bazy danych.

Dzięki zastosowanym technologią, aplikacja będzie mogła trafić 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.

Środowisko produkcyjne.

Do stworzenie części praktycznej tej pracy autor użył następujących narzędzi przy pisaniu pracy.

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

Tworząc pracę autor wybrał ten program ze względu na:

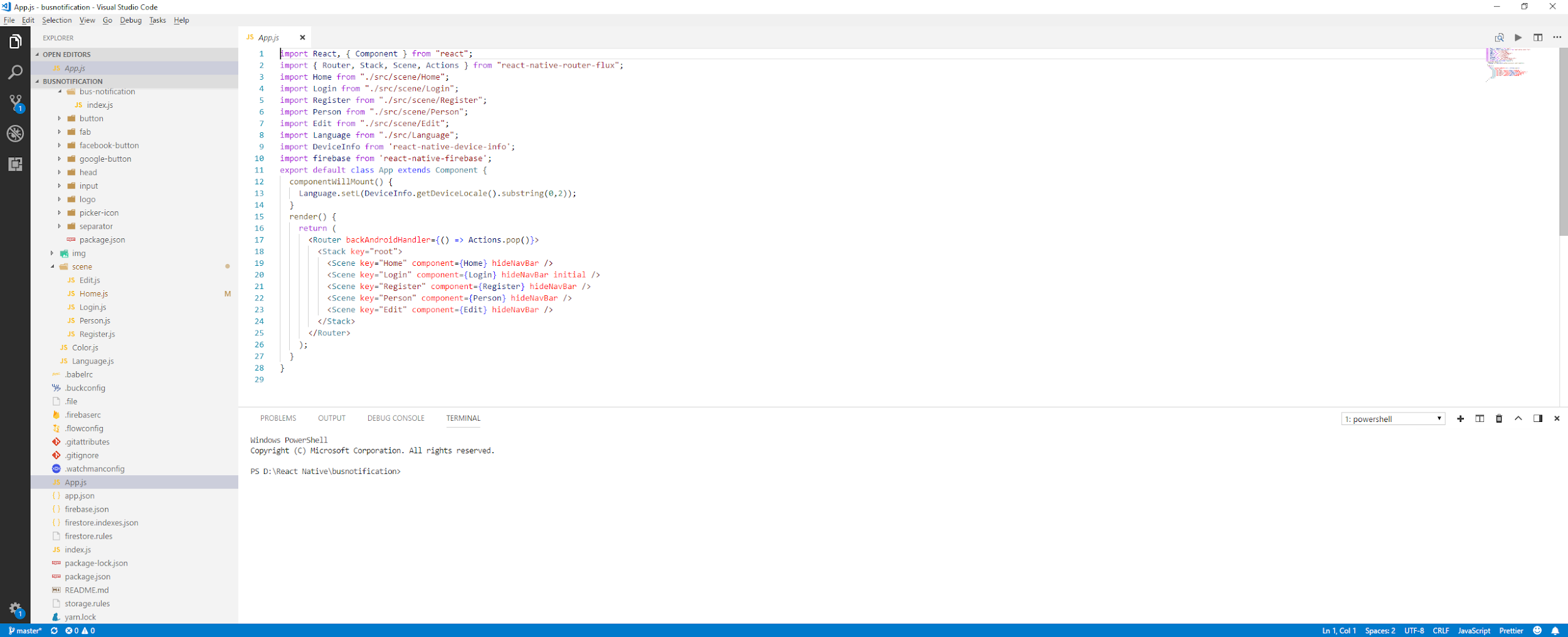
· Szybkość działania

· Dużą ilość dodatków ułatwiających pisanie

· Przejrzystość interfejsu

· Jest darmowy

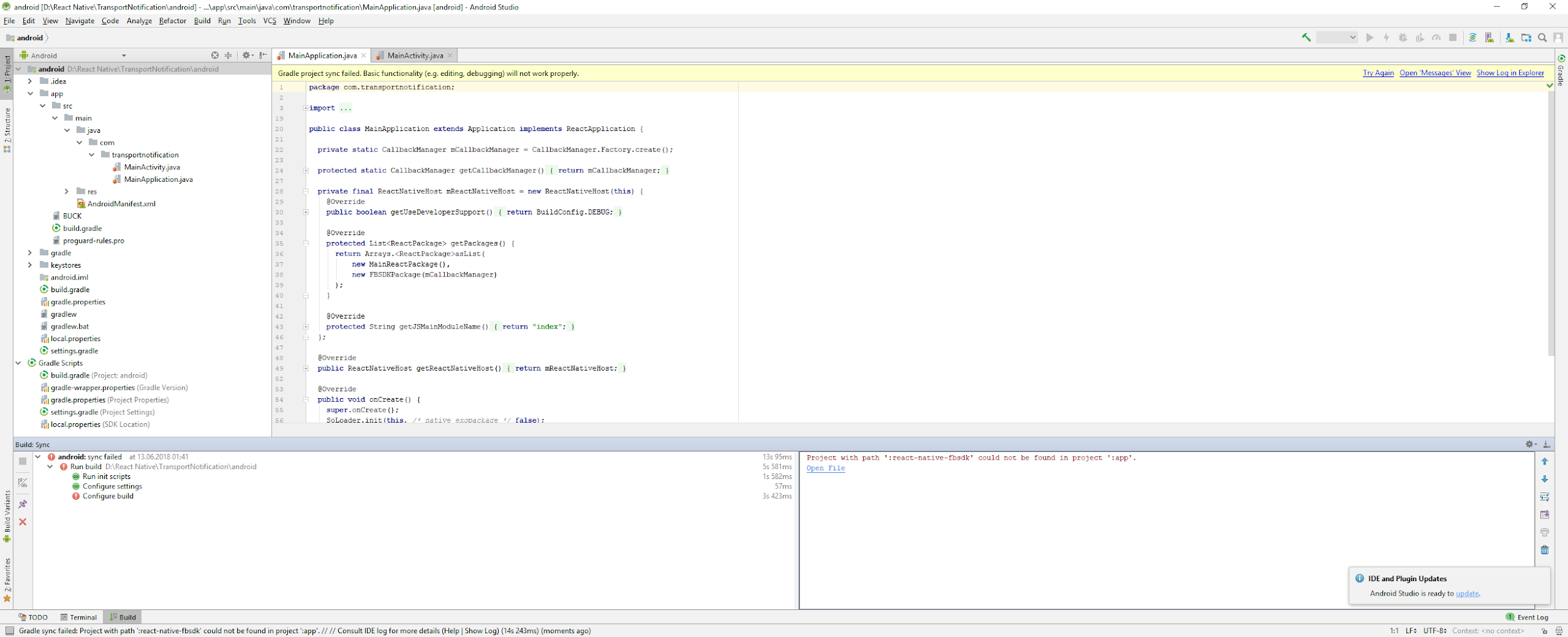
· Można w nim pracować na wielu platformach, co pozwala na w łatwy sposób dopasowywać aplikację pod telefony z systemów IOS, od firmy Apple.



Rys. 1. Program Visual Studio Code.

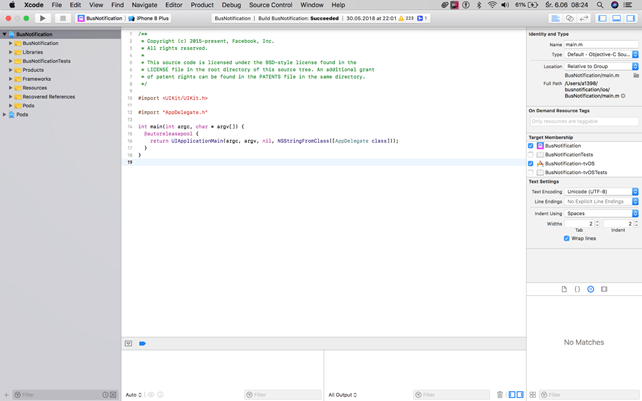
React Native Tools jest to darmowy dodatek do Visual Studio Code pozwalający na pisanie aplikacji w języku React Native w Visual Studio Code. Autor wybrał ten dodatek, ze względu na to że jest to oficjalny dodatek do Visual Studio Code do programowania aplikacji w React Native

Android Studio jest to darmowy edytor kodu platformy Android, pozwalający na tworzenie aplikacji w języku Java oraz Kotlin. Autor użył tego programu do łączenia komponentów napisanych w języku Java na system Android z React Native.



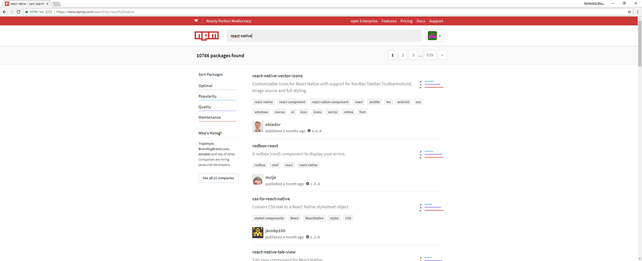
Rys. 2. Program Android Studio

XCode jest to darmowy edytor kodu platformy IOS, pozwalający na tworzenie aplikacji w języku Swift oraz Objective C. Autor użył tego programu do łączenia komponentów napisanych w języku Objective C na system IOS z React Native.



Rys. 3. 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.

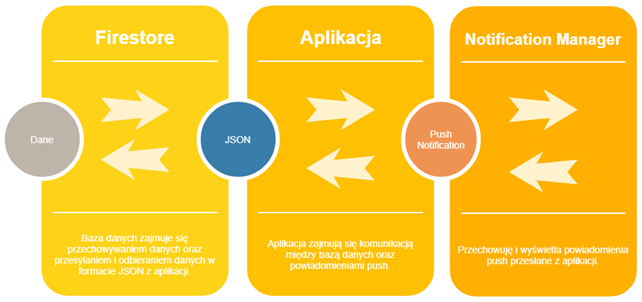


Rys. 4. Katalog pakietów na stronie npmjs.com.

2. Project Aplikacji

##### 2.1 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 Firestorm, a Notification Managerem, gdzie znajdują się powiadomienia.



Rys. 5. Przepływ danych w aplikacji Bus Notification.

2.2 Widoki

Widoki aplikacji są to poszczególne okna które wyświetlają zawartość aplikacji. Aplikacja będzie powinna się składać z co najmniej 5 widoków:

Widok logowania, za pomocą, którego można się będzie zalogować do aplikacji. W widoku powinna się znajdować się nazwa aplikacji pole do wpisanie emailu, pole do wpisaniu hasła i przycisk zaloguj, powinien się zajmować przycisk do przejścia do rejestracji.

Widok rejestracji za pomocą, którego można się będzie zarejestrować do aplikacji.W widoku powinna się znajdować się nazwa aplikacji pole do wpisanie emailu, pole do wpisaniu hasła i powtórnego wpisania hasła w celu weryfikacji podanego hasła oraz przycisk zarejestruj.

Widok listy godzin, gdzie przejrzyście będą pokazane godziny, w których pojawią się przypomnienia o wyjściu kurs autobusu. Lista powinna być taka przedstawiona, żeby użytkownik nie musiał wchodzić w szczegóły, aby sprawdzić co znajduje się w niej elemencie. Użytkownik powinien z widoku listy móc dodać przypomnienie, usunąć przypomnienie, aktywować przypomnienie oraz przejść do edycji przypomnienia.

Widok dodawania i edycji powiadomienia to widok, który pozwoli na dodawanie i edycję elementu z listy powiadomień. Widok powinien zawierać pole do wpisania środka transportu, kierunku podróży ,godziny oraz przycisk do potwierdzenia zapisu danych.

Widok wylogowania i zmiany hasła to widok, który ma pozwolić na wylogowanie z aplikacji oraz zmianę hasła. Powinny się w nim znajdować przycisk wyloguj, pola do podania hasła i powtórzenia, w celu weryfikacji oraz zapisu nowych danych.

2.3 Powiadomienia

Powiadomienia są to małe widoki ukazujące się na ekranie telefonu przez krótką chwilę nad inną aplikacją, które są później przenoszone do centrum powiadomień. Z powiadomieniami można wejść w interakcję.

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.

2. Założenia aplikacji.

Główne założenia aplikacji Bus Notification:

· Prostota użytkowania

· Szybkość działania

· Wieloplatformowość

· Możliwość obsługi w kilku językach

· Łatwy sposób tworzenia konta w aplikacji

· Nowoczesny i uniwersalny interfejs aplikacji

4. Funkcje Programu

##### Założenia funkcjonalne:

##### 1. Logowanie za pomocą mailu, konta Facebook oraz konta Gmail za pomocą Firebase Auth.Logowanie powinno by się odbywać pomocą emailu oraz hasła. Przetwarzaniem podanych przez użytkownika danych będzie zajmować się

##### 

##### 2. Rejestracja za pomocą mailu i hasła

##### 3. Synchronizacja danych pomiędzy bazą danych a aplikacją

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

##### 5. Możliwość zmiany częstotliwości powiadomień

##### 6. Kilka języków obsługiwania angielski, polski

##### 7. 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.

5. 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 podstawie 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.

a. Środowisko produkcyjne.

3. Analiza wymagań oraz projekt aplikacji.

4. Założenia aplikacji.

5. Funkcje programu.

b. Ogólne wymagania funkcjonalne.

c. Ogólne wymagania niefunkcjonalne.

6. Projekt aplikacji:

d. Baza danych.

e. Opis tabel.

1. Javascript

2. React Native

3. CSS

4. Npm

5. Jsx

6. State i Props

7. Różnice między IOS i Android

8. Komponenty natywne

9. Komponenty React Native

10. Tworzenie Komponentów

##### 

##### 1.

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 lub 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 przyspieszyły 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 przyspieszyć 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 publikowania oraz pobierania pakietów z bazy. Za pomocą strony mamy dostęp do bazy z pakietami.

“Recykling kodu” Reusable-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.