1. Motywacja

Temat pracy dotyczy nowych technologii, w szczególności smartfonów oraz oprogramowania jakim operują. Motywacją do jego poruszenia jest zainteresowanie Autora przestrzenią użytkową platform i aplikacji.

Rynek smartfonów i innych smart-urządzeń stale się rozrasta. Większość ludzi do przeglądania zasobów Internetu oraz wzajemnej komunikacji, wykorzystuje smartfony. Dlatego, biorąc pod uwagę obecną sytuację, tworzenie aplikacji na tego typu urządzenia jest bardzo opłacalne. Aplikacja mobilna ma dwie kluczowe cechy: jest łatwo dostępna i szybko dociera do szerokiego grona potencjalnych użytkowników – związane jest to z szerokimi możliwościami wykorzystania bazy aplikacji, czyli sklepu, którego posiadaczami są wszyscy użytkownicy smartfonów. Za jego pomocą można szybko pobrać (w przypadku aplikacji darmowych) lub zakupić wybraną aplikację.

Poprzez analizę sytuacji rynkowej i zapoznanie się z potrzebami społeczeństwa, Autor postanowił skupić się na usprawnieniu relacji komunikacyjnej, w obrębie transportu publicznego. Będąc świadkiem, a także uczestnikiem takich sytuacji jak: spóźnienia na autobus, spowodowane, zbyt późnym wyjściem z domu lub roztargnieniem, Autor wymyślił aplikację, której celem jest zminimalizowanie takiego zjawiska. Rolą aplikacji jest przypomnienie ludziom, że już powinni wyjść na autobus, pociąg itd.

2. .Cel.

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

Praca przedstawia nowy sposób pisania hybrydowych aplikacji mobilnych, tworzonych w języku Javascript, a także wykorzystania platformy Firebase do autoryzacji użytkowników i bazy danych oraz testowania aplikacji.

React Native to 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 trzeba zapłacić.

Dzięki zastosowanym technologiom, aplikacja będzie mogła trafić i być używana przez większość użytkowników systemów mobilnych (Android + IOS 99,8%).

Aplikacja powinna być:

1.intuicyjna w taki sposób, aby każdy, mimo braku znajomości języka polskiego czy angielskiego, mógł ją obsługiwać.

2.przygotowana w technologii, która pozwala na jednoczesne tworzenie aplikacji na system Android i IOS.

3.przejrzysta i przyjemna w obsłudze dla użytkownika.

Interfejs powinien być tak zaprojektowany, aby niezależnie od platformy czy wielkości wyświetlacza, zapewniać wygodę w użytkowaniu.

Powiadomienia przypominające o wyjściu na autobus, pociąg itd. powinny się pojawiać niezależnie czy aplikacja jest uruchomiona, czy wyłączona. Jednocześnie, połączenie z Internetem, nie może wpływać na jakość usługi.

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

3. Środowisko produkcyjne.

Do stworzenie części praktycznej tej pracy, Autor użył m.in. Visual Studio Code. Jest to darmowy edytor kodu, działający na systemach Windows, Linux i MacOS.

Tworząc pracę, Autor wybrał ten program ze względu na:

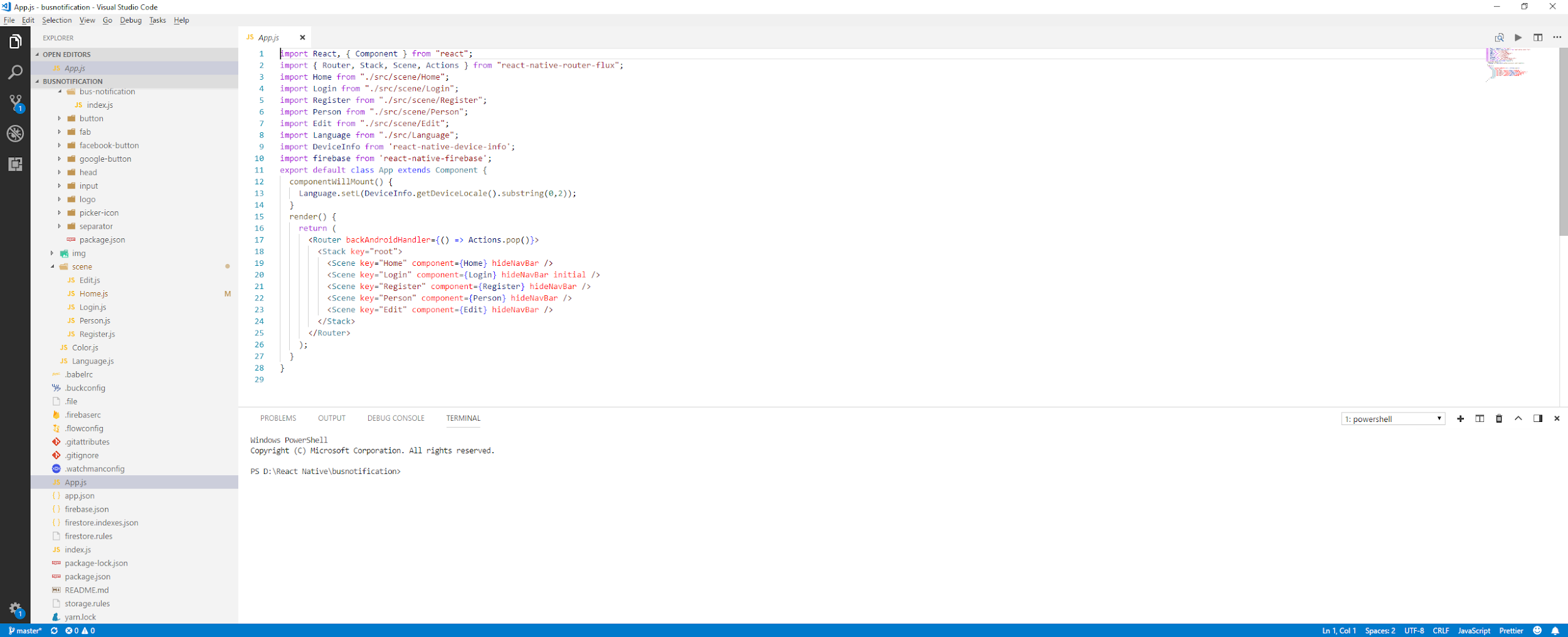
· Szybkość działania

· Dużą ilość dodatków ułatwiających pisanie

· Przejrzystość interfejsu

· Bezpłatną opcję użytkowania

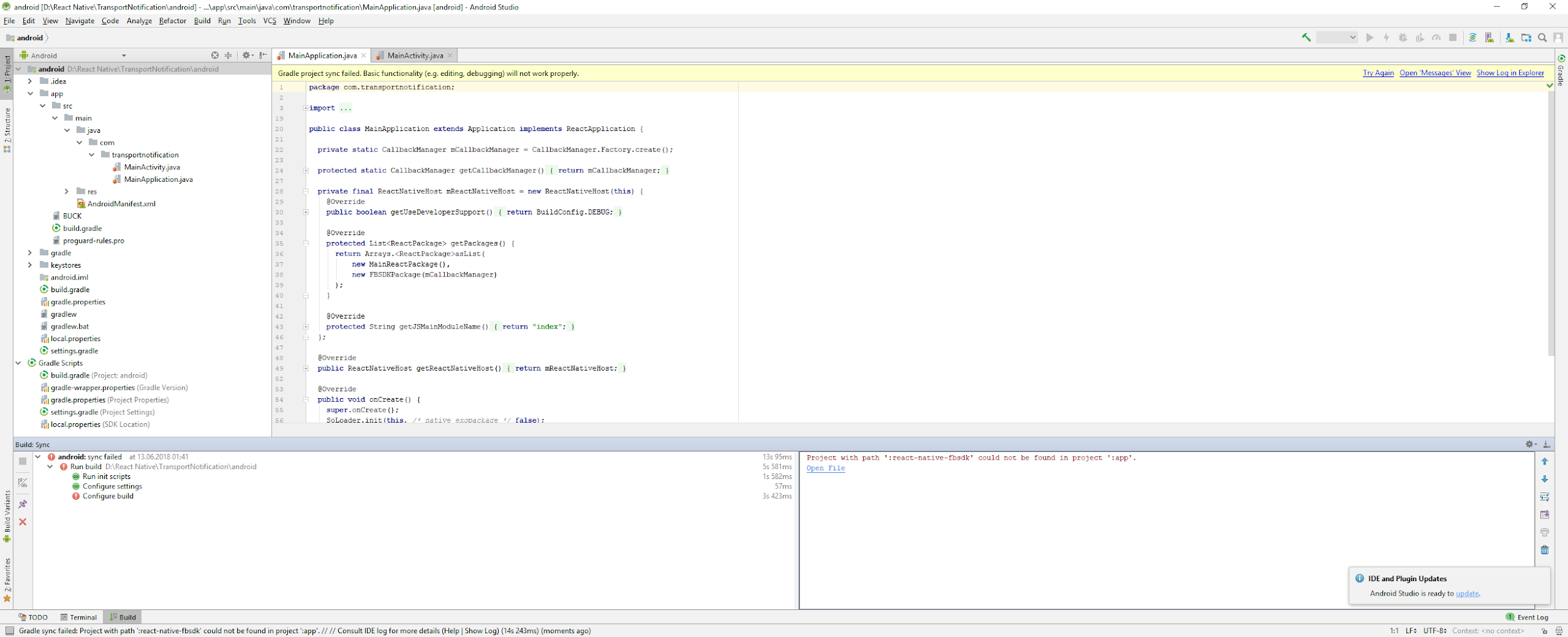
· Możliwość pracy na wielu platformach, co pozwala w łatwy sposób dopasowywać aplikację pod telefony z systemów IOS, od firmy Apple.



Rys. 1. Program Visual Studio Code.

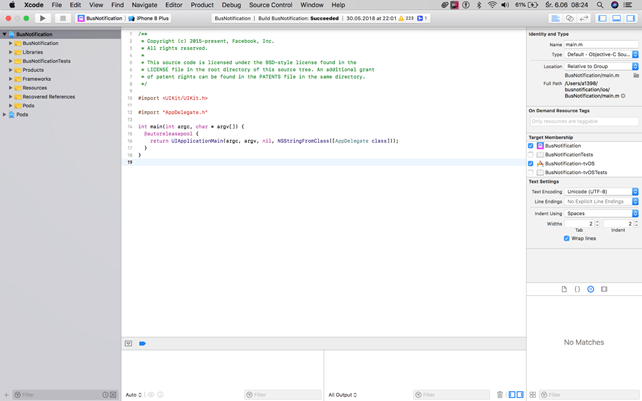
React Native Tools jest to darmowy dodatek do Visual Studio Code pozwalający na pisanie aplikacji w języku React Native w Visual Studio Code. Autor wybrał ten dodatek, ze względu na to, że jest to oficjalny dodatek do Visual Studio Code do programowania aplikacji w React Native

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.



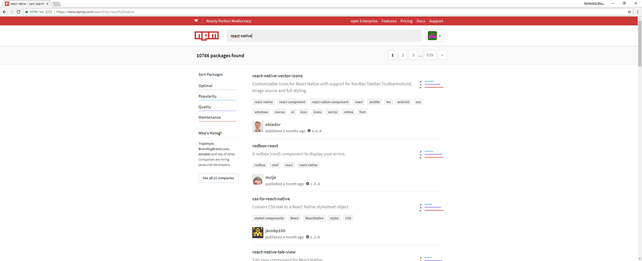
Rys. 2. Program Android Studio

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 ze strony npmjs.com, wyświetlające listę pakietów, baza pakietów dla języka Javascript i frameworków. Opiera się ono o język JavaScript oraz klient na systemy z Windows, Linux i MacOS i pozwala na zarządzanie pakietami. Npm ułatwia tworzenie kodu, ponieważ daje łatwy dostęp do ogromnej liczby darmowych pakietów, z których można skorzystać, przy tworzeniu aplikacji.

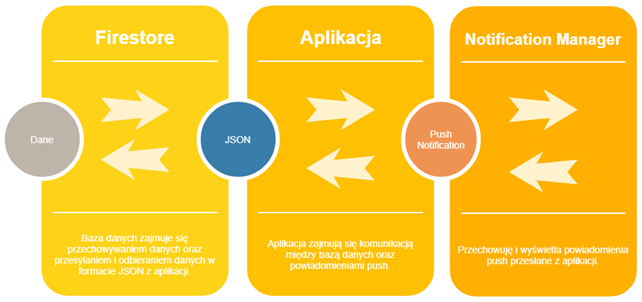


Rys. 4. Katalog pakietów na stronie npmjs.com.

4. Project Aplikacji

##### 4.1 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.Pobrane pliki komunikuje między Firestorm, a Notification Managerem, gdzie znajdują się powiadomienia.



Rys. 5. Przepływ danych w aplikacji Bus Notification.

4.2 Widoki

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

Widok logowania, za pomocą którego można się będzie zalogować do aplikacji. Ważne, aby w widoku znajdowała się nazwa aplikacji, pole do wpisanie emailu, pole do wpisaniu hasła i przycisk zaloguj, a także przycisk przejścia do rejestracji.

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

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

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, a także 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 to małe widoki, ukazujące się na ekranie telefonu przez krótką chwilę. Usytuowane są nad inną aplikacją. Przenoszone są później do centrum powiadomień. Z powiadomieniami można wejść w interakcję lub za pomocą kliknięcia w nie wejść do aplikacji.

W aplikacji powiadomienia powinny działać w taki sposób, aby użytkownik otrzymywał je codziennie o wskazanej przez siebie godzinie, nawet jeśli aplikacja nie będzie uruchomiona i telefon nie będzie połączony z internetem.

Powiadomienie powinno być czytelne i zawierać ikonę aplikacja wraz z nazwą, kierunek trasy, rodzaj transportu i godzinę.

5. Założenia aplikacji.

5.1 Główne założenia aplikacji Bus Notification.

Aplikacja powinna być na tyle prosta, aby użytkownik z podstawową znajomością języka angielskiego, mógł sobie poradzić z obsługą. Istotna jest także przejrzystość aplikacji, tzn to że użytkownik bez trudu może bez trudu określić co znajduje się na ekranie użytkownika i do czego służy, co pozwala na wygodną i szybką obsługę.

Oprócz szybkiej obsługi aplikacja powinna sama szybko działać pod względem animacji czy zapisywania danych. Nie jest dopuszczalne, aby aplikacja się zacinała czy pokazała pusty ekran. W momencie pobierania danych powinien pojawić się komponent z paskiem ładowania albo stosowny komunikat.

Największą zaletą React Native jest multiplatformowość, tzn możliwość tworzenia jednego kodu programu w celu uruchomienia na innych platformach. Domyślnie RN umożliwia tworzenia aplikacji na platformę Android oraz IOS, ale są także rozszerzenia, która pozwalają na uruchomienie tego samego kodu z małymi zmianami na innych platformach. Wieloplatformowość jest ważna z tego względu, że pozwala zwiększyć liczbę potencjalnych użytkowników aplikacji.

Kolejną rzeczą zwiększającą liczbę teoretycznych użytkowników jest możliwość obsługi w kilku językach. Wiele popularnych aplikacja pozwala na obsługę w wybranym języku. Poprawia to dostępność aplikacji dla nowych użytkowników, a co za tym idzie wygodę korzystania z aplikacji. Nie się stworzyć aplikacji bez napisów, a dodanie napisów w kilku językach, może pomóc w rozwoju aplikacji i dotarcie do większej ilości użytkowników.

Bazę użytkowników może także zwiększyć integracja z znanymi serwisami społecznościowymi takimi jak Facebook , czy z kontem Google. Wiele nowoczesnych aplikacja pozwala na logowanie się za pomocą konta na portalu Facebook, czy dodanie posta z aplikacji. Taka integracja znacząco przyspieszy utworzenie konta, a co za tym idzie ułatwi korzystanie z aplikacji użytkownikowi.

Nowoczesny i intuicyjny interfejs jest równie ważny jak i warstwa aplikacji związana z programowaniem, często nawet wygląd aplikacja przesądza o jej sukcesie. Ważny jest aby aplikacja miała nowoczesny i spójny interfejs, tzn przed tworzeniem nowych własnych komponentów w aplikacji powinniśmy się zastanowić nad użyciem już istniejących, Każda platforma ma swój schemat tworzenia interfejsu. Autor pracy zdecydował się na tworzenie według Material Design.

Jest to schemat tworzenia aplikacji stworzony przez firmę Google, bardzo często używany przy tworzeniu aplikacji na platformę Android. Tworząc aplikację multi platformą należy pamiętać, aby interfejs użytkownika był tak zaprojektowany, żeby działał tak samo na wszystkich platformach.

Odpowiedni projekt aplikacji może oszczędzić czas na przygotowanie osobnych komponentów dla różnych platform. Należy pamiętać o różnych w obsłudze między platformami. Autor pracy przyjął, że aplikacja będzie kompatybilna z telefonami z systemami Google i Apple. Korzystanie z tych systemów się znacząco różni, głównie ze względu na przyciski.

Telefony z Androidem są wyposażone w 3 przyciski cofanie, który cofa użytkownika do poprzedniego okna , “domowy”, przenoszący na pulpit użytkownika oraz przycisk ostatnich używanych aplikacji, natomiast IPhone ma jeden przycisk “domowy”, którego pojedyncze kliknięcie przenosi na pulpit, a podwójne otwiera ostatnio używane aplikacje. W związku z brakiem przycisku cofania aplikacja powinna mieć przycisk cofania, najczęściej znajduję się on w górnym lewym rogu i ma symbol strzałki w lewo.

Druga sprawą jest kompatybilność aplikacji z telefonami na sklepie Google nakłada obowiązek z kompatybilności z odpowiednią wersją systemu, natomiast sklep Apple nakłada na więcej obowiązków. Aplikacje powinny być przygotowane pod wszystkie IPhony dostępne w sprzedaży. Najnowszym IPhonem jest IPhone X, charakteryzującym się dużym ekranem z wycięciem na u góry ekranu. Należy na to zwrócić uwagę przy projektowaniu interfejsu, ze względu na żeby treści wyświetlały się poprawnie mimo wcięcia.

6. Funkcje Programu

##### 6.1 Założenia funkcjonalne.

##### Logowanie i rejestracja w aplikacji powinna być realizowana za pomocą Firebase Authentication, które pozwala w łatwy sposób zarządzać między innymi użytkownikami i metodami logowania. Pozwala to sprawdzić liczbę użytkowników aplikacji z jakich lokalizacji się rejestrują oraz skonfigurować im metody logowania i rejestracji, co ułatwi i przyśpieszy ten proces.

##### Google Firebase oferuje dwa rodzaje baz danych, Realtime Database i Cloud Firestore.

##### Realtime Database ma strukturę drzewiastą i jest oparta o format JSON, tzn każdy element może się rozgałęziać i mieć elementy podrzędne jak korzenie drzewa. JSON jest to popularny format zapisu danych stosowany do przedstawiania danych, nie zapisujemy w nim typów przechowywanych danych tylko informacje o nazwach pól klasy i obiektach.

##### Cloud Firestore jest oparta na kolekcjach i dokumentach, kolekcja to zbiór dokumentów, dokument to jednostka danych ma swoje pola które mogą zawierać dane. Dane w Firestore mają swój typ, co pozwala nam łatwiej ocenić z jakimi danymi mamy do czynienia. Występują tu także referencję do dokumentu, tzn można się odwołać do danych znajdujących się w innym dokumencie czy w innej kolekcji.

Autor aplikacji wybrał Cloud Firestore, ze względu typowane zmienne, referencje i strukturę bazy, która ułatwia pracę z nią.

Aplikacja potrafi w różny sposób odbierać i przesyłać powiadomienia. Jednym z tych rodzajów są powiadomienia push, tzn powiadomienia, które są można pokazywać o danych godzinie i wyświetlać je cyklicznie. Nie są zależne od połączenia z internetem czy włączonej aplikacji.

Pozwoli to na że użytkownik mimo tego, że aplikacja jest wyłączona, będzie nadal powiadamiany o tym, że powinien już wyjść na autobus.

##### Aplikacja powinna wyświetlać informacje w kilku językach. Autor skupił się na dwóch polskim i angielskim. Sposób dodawania nowych języków powinien być prosty i nie ingerować w powstały już kod. Aplikacja powinna dopasowywać język do tego jaki został ustawiony, w telefonie to pozwoli zmienić język, kiedy użytkownik zmieni domyślny język w telefonie.

6.2 Wymagania niefunkcjonalne.

Aplikacja będzie działać w czasie rzeczywistym, tzn. zmiany dokonane przez użytkownika, będą od razu zapisywane w bazie danych, tzn dodanie, zmiana lub usunięcie elementu powinny być natychmiast przesyłane do bazy. Dzięki temu użytkownikowi lepiej wygodniej korzysta się z aplikacji.

Skomplikowana i wolno działająca aplikacja nie szans na dotarcie do dużej ilości użytkowników. Autor powinien się skupić na tym, że aplikacja działała płynnie i korzystanie z niej było jak najprostsze.

Nowoczesny i atrakcyjny wygląd może być kluczem do sukcesu przy tworzeniu aplikacji. Mimo tego że jest wiele podobnych aplikacji, to najwięcej osób korzysta z tych, które oprócz dużej liczby funkcji charakteryzują się przejrzystym i cieszącym oko użytkownika interfejsem.

Tworząc teraz aplikację należy wziąć pod uwagę kilka platform. React Native pozwala na tworzenie aplikacji multiplatformowej, ale należy pamiętać o tym przy projektowaniu interfejsu, aby był dopasowany do obu platform jednocześnie i pozwalał na wygodną obsługę niezależnie od platformy.

Jedną z największych zalet React Native jest to można w łatwy sposób rozbudowywać aplikację. Aplikacja składa się z widoków, na których znajdują się komponenty, są to przyciski czy pola tekstowe, a także inne elementy służące do manipulowania danymi oraz aplikacją. Poprzez strukturę aplikacji pisanej w React można w łatwy sposób dodać czy wykonać nowy komponent.

RN korzysta z JSX, jest to rozwinięcie składni języka JavaScript przypominające XML lub HTML.

Aplikacja powinna mieć możliwość łatwego przesyłania błędów i raportowania ich, Taką funkcjonalność oferuje Firebase Crashlytics. Agreguje ona wszystkie błędy wysłane przez aplikację i dzieli na typy . Pozwala na postawie otrzymanych danych w łatwy sposób odtworzyć błąd na innym urządzeniu, albo sprawdzić na jakich urządzeniach występują podobne błędy.

7. Project Bazy.

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 pliki użytkowników wybrał Cloud Firestore. Jest to baza danych oparta o NoSQL. Jest bazą, która opiera się na 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 informujące o tym, 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 swój uid, pozwalający na wyróżnienie go na liście z dokumentami.

**TEGO NIE BIERZ POD UWAGĘ TO NOTATKI DLA MNIE**

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. Znajduje 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 to zbiór funkcji albo komponentów, ułatwiający pisanie oprogramowania. Darmowe konto na npmjs.com pozwala na tworzenie tylko publicznych pakietów. Wbudowana wyszukiwarka ułatwia wyszukiwanie najlepiej ocenianych pakietów. Npm używałem (albo całą pracę AUTOR UŻYWAŁ, albo UŻYWAŁEM) 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 ze strony, bazy z pakietami do pobrania oraz klienta npm. Klient służy do publikowania oraz pobierania pakietów z bazy. Strona umożliwia dostęp do bazy z pakietami.

“Recykling kodu” Reusable-code

Serwis wpisuje się w modułowy sposób tworzenia aplikacji. Tworząc aplikację, piszemy komponenty czy funkcje, które można powtórnie wykorzystać w innym programie. Npm pozwala w łatwy sposób udostępniać swoje oprogramowanie. Organizacja serwisu pozwala na stworzenie własnego konta oraz organizacji, w której będziemy udostępniać swoje komponenty, a także na prywatne publikowanie pakietów, (są wtedy dostępne tylko dla twórcy), bądź organizacji, w której zostały udostępnione. Niestety za tę 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 także podczas tworzenia aplikacji mobilnych czy aplikacji na komputery z systemem windows, mac os, linux.

JAK SKOŃCZYSZ PAMIĘTAJ O UJEDNOLICENIU PRACY. JEŚLI ROBISZ WYLICZENIE OD NUMERÓW TO DALEJ TEŻ OD NUMERÓW, A NIE OD PAUZ. PRZCISKI TYPU PUSH, PASUJE DODAĆ W CUDZYSŁOWIE. AHOJ PRZYGODO!