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.

1. Motywacja.

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.

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. 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

##### Założenia funkcjonalne

##### Aplikacja ma pozwalać na utworzenie konta w łatwy sposób, dodanie informacji kiedy mają się pojawiać przypomnienia o transporcie. Aplikacja ma być w kilku językach i działać na plafotmach Android i IOS.

##### Lista dostępnych funkcji:

##### 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, niemiecki, polski

##### Aplikacja działająca na Android i IOS

##### Utworzenie reklam za pomocą Admob od Googleimage14.png

##### 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.

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.