**Paśtwowa Wyższa Szkoła Zawodowa w Tarnowie**



Programowanie systemów mobilnych II

Projekt

**Prowadzący:**

mgr inż. Tomasz Gryl

**Kierunek:**

Informatyka, rok 2018/2019

**Autorzy:**

Kamil Tomasiewicz

Krzysztof Szynal

1. Podstawowe informacje

**Temat**: Wielozadaniowy harmonogram

**Cel Projektu:** Celem projektu jest napisanie aplikacji mobilnej na platforme Android z użyciem frameworka React-Native w ktorej zadaniem jest stworzenie kalendarza który bedzie miał możliwość przypomnienia o nadchodzącym wydarzeniu oraz funkcją notatnika.

**Użyte biblioteki:**

React-Native-Navigation by wix,

React-Native-elements,

React-Native-swipe gestures,

React-Native-vector-icons,

React-Native-sqlite-storage,

React-Native-splash-screen,

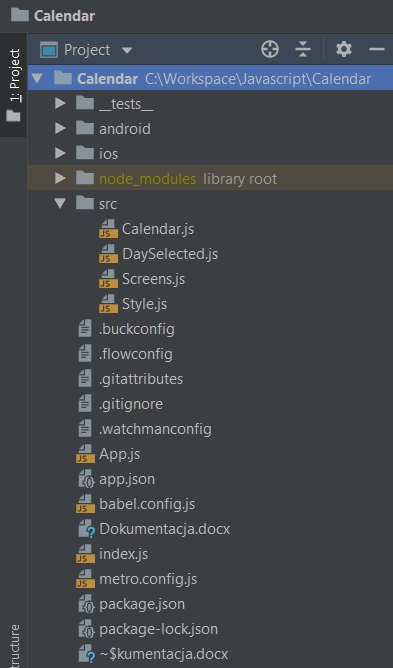
React-Native-push-notifications

**Sposób instalacji:** W Pierwszej kolejności należy utworzyć projekt poprzez użycie konsoli

***react-native init Calendar*** a w nastepnej kolejności zainstalować ww. Biblioteki. (Sposoby instalacji są na stronie internetowej bibliotek)

1. Specyfikacja programu

**Struktura plików**

****

**Katalogi**

**android / ios -** W tych katalogach zostały zawarte informacje jak aplikacja ma zostać uruchomiona zależnie od systemu operacyjnego.

**node\_modules** – W danym katalogu znajdują sie wszystkie biblioteki zainstalowane od *”npm”* na potrzeby dzialania programu

**src –** Katalog zawierajacy informacje z wyglądem oraz o działalności poszczególnych ekranow

**Pliki**

**Calendar.js -**  Widok kalendarza wraz z zawartymi metodami

**DaySelected.js –** Widok notatnika w wybranym dniu

**Style.js** – Plik ze wszystkimi stylami użytymi w aplikacji.

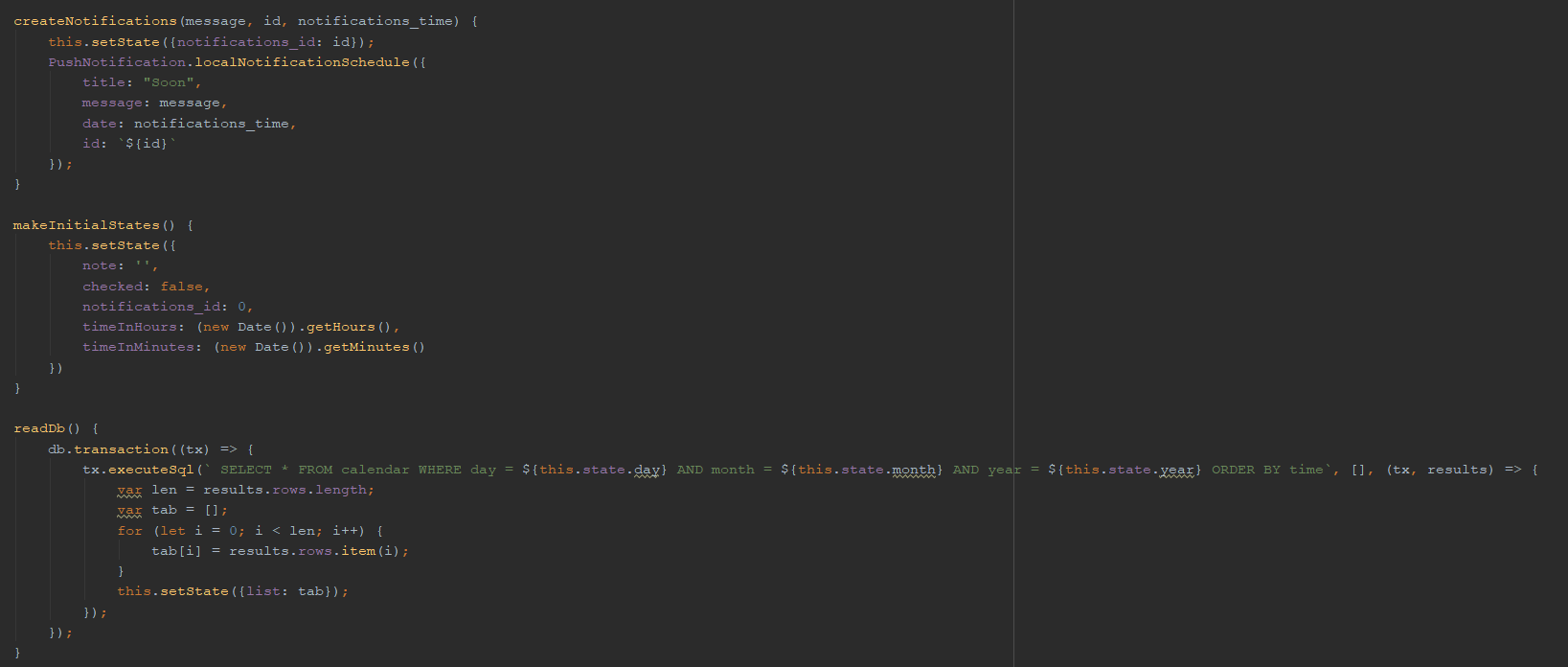
**Screens.js** – Inicjalizacja ekranów (Nadanie nazw ekranom).

**Index.js** – Konstrukcja aplikacji oraz ogólne ustawienia ekranów.

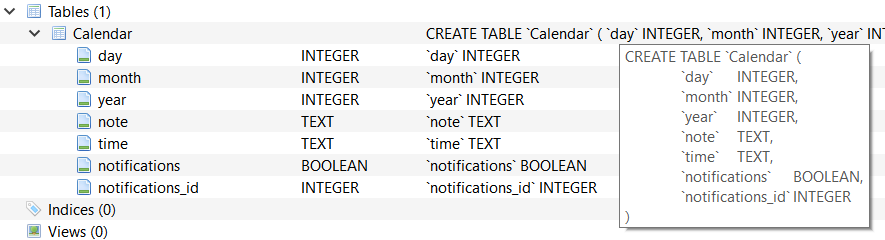
**package.json** - Lista zainstalowanych bibliotek używanych przez aplikacje.

**app.json** – Plik z danymi o nazwie wyświetlanej aplikacji.

**Kod**

****

**Struktura tabeli calendar SQL**

****

**State**

**This.state** – Sa to stany które sie delkaruje w construktorze, a nastepnie można wczytywać albo modyfikować wartość danych stanow poprzez odwołanie sie bezposrednio do nich. Aby użyć wartości za state należy sie odwołać w sposób: **this.state.variable,** ajeżeli zmodyfikować stan należy posłużyć sie **this.setState({variable: dane})**.

**Funkcje**

Wymienimy tylko garstke funckji zawartych w projekcie

**readDb** – Odpowiada za wczytanie danych z bazy danych oraz przpisanie danych do stanu *list* co skutkuje, że component FlatList zacznie sie renderować raz jeszcze gdyż stan listy podanej do componentu uległ zmianie

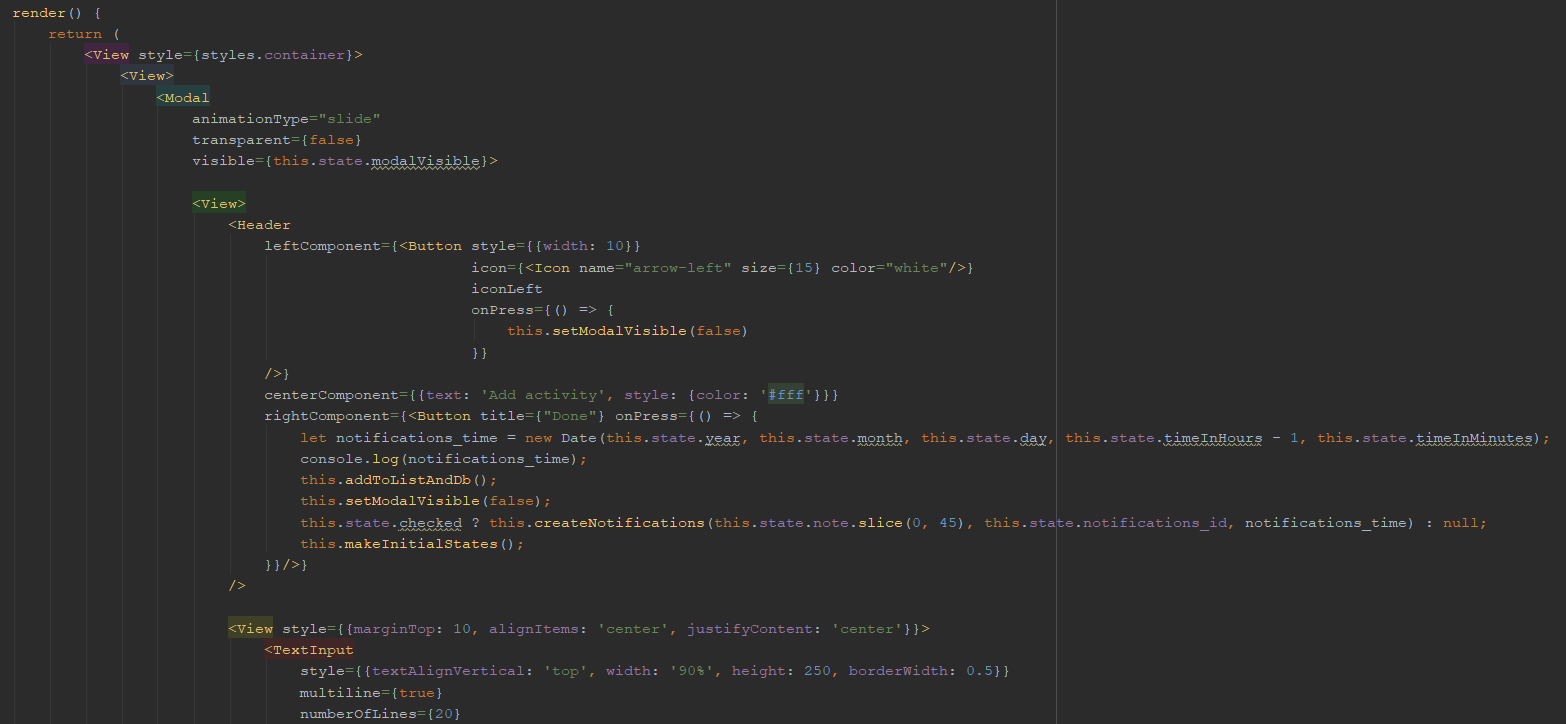
**componentDidMount** – Ważne jest, że componentDidMount jest uruchamiany zanim część androida i react-native jest zdolna do pracy i na potrzeby projektu wczytuje baze danych oraz wyłącza splash-screen

**renderActivity** – Podana została w FlatLiscie by kazdy element w liscie został wyrenderowany w sposób zawarty w podanej funckji

**createNotifications** – Funkcja odpowiada za utworzenie notifikacji o odpowiedniej godzinie, dniu, miesiacu oraz roku

**Render**

Częśc renderowa jest dzięki React-Native i pozwala na testowanie w szybkim tępie oraz pisanie aplikacji mobilnych



**Componenty**

Miedzy innymi zostały użyte componenty takie jak:

**View** - component podstawowy, który robi za fundament

**Modal –** Otwiera modal czyli tzw. okno

**TouchableOpacity –** Przycisk który ‘błysnie jak zostanie wciśniety’

**Button –** Podstawowy przycisk, wymaganymi propsami by został wyrenderowany sa title oraz onPress

**FlatList –** Lista, która wymaga by podane zostały propsy takie jak *data* oraz *renderItem*

**Text Input –** Pole do wpisywania tekstu

**Screeny**

