**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Проект з дисципліни

**«Розробка програмного забезпечення під мобільні платформи»**

на тему:

**«Розробка додатка для пошуку ресторанів за гео-позицією»**

**Виконав**:

студент 4 курсу ОС «Бакалавр»

спеціальності «Компʼютерні науки»,

освітньої програми «Інформатика»,

групи ТТП-42

*Маєвський Любомир Андрійович*

**Київ-2022**

ЗМІСТ

[ОПИС ЗАВДАННЯ](#_gjdgxs) 3

[КОРОТКИЙ ВИКЛАД ТЕОРІЇ](#_30j0zll) 4

[ОПИС РОБОТИ З ПРОГРАМОЮ ТА ЗНІМКИ ЕКРАНУ](#_3znysh7) 7

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ](#_2et92p0) 11

[ДОДАТКИ](#_tyjcwt) 12

[Додаток А](#_3dy6vkm) Фрагмент Коду, Головна сторінка 12

[Додаток Б](#_1t3h5sf) Фрагмент Коду, Пошук Ресторанів 13

[Додаток В](#_4d34og8) Фрагмент Коду, Відображення Ресторанів 15

[Додаток Г](#_2s8eyo1) Фрагмент Коду, Збереження Улюблених Ресторанів 16

# ОПИС ЗАВДАННЯ

Завдання полягає у розробці додатку, яку якому буде можливість пошуку ресторанів за їх гео-позицію:

1. Ввід:

* Назва геопозиції

1. Вивід:

* Список ресторанів
* Кожен ресторан має інформацію про меню та заклад

1. Робота з файлами:

* Можливіть відмітити будь-який ресторан як улюблений
* Збереження улюблених ресторанів

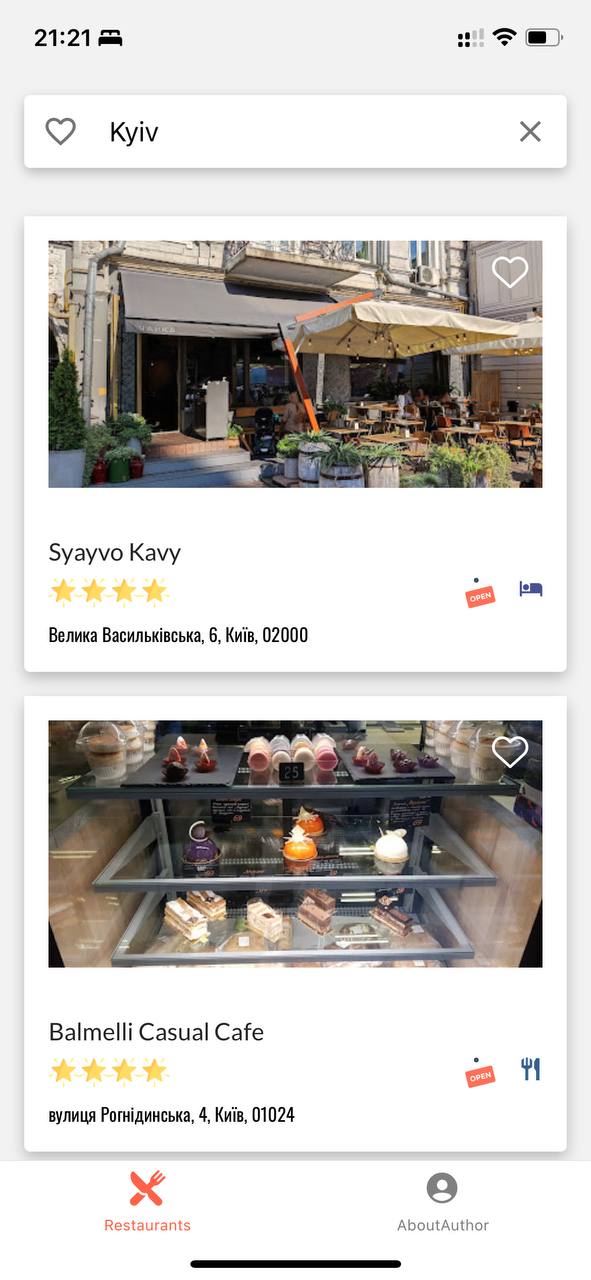
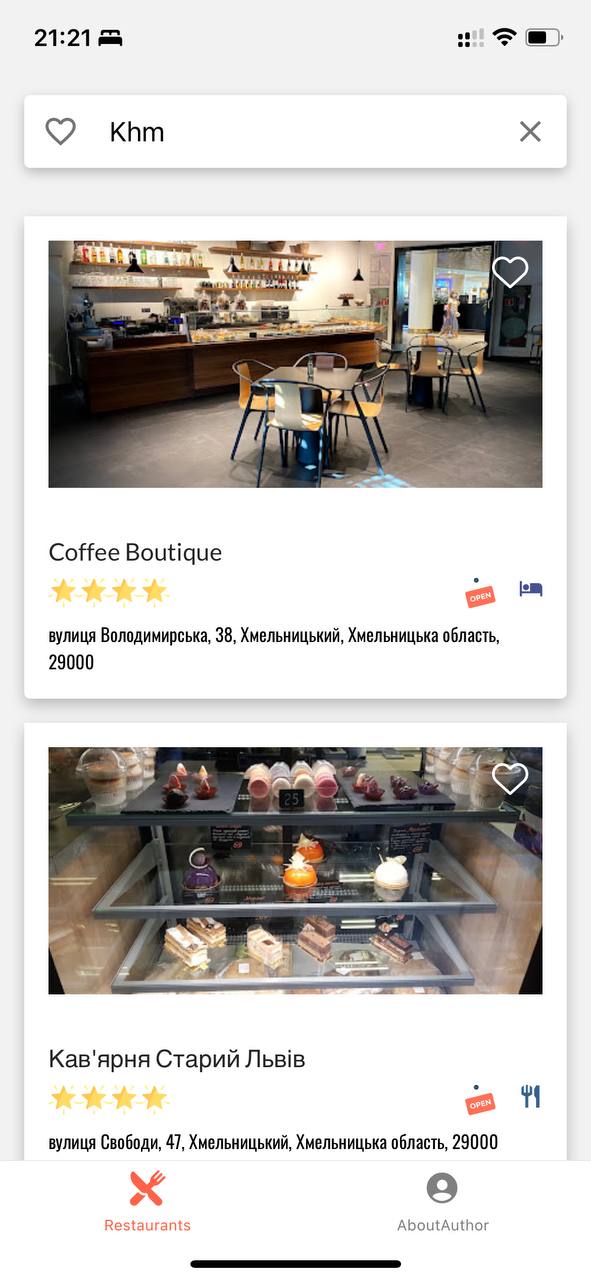
# КОРОТКИЙ ВИКЛАД ТЕОРІЇ

Для реалізації даної задачі була використана бібліотека React Native. React Native — це фреймворк JavaScript з відкритим вихідним кодом, розроблений для створення додатків на кількох платформах, таких як iOS, Android, а також веб-додатків, використовуючи ту саму базу коду. Він заснований на React і є досить популярним засобом для розробки мбільних додатків. React Native використовує JavaScript для компіляції користувальницького інтерфейсу програми, але з використанням нативних переглядів ОС. Для більш складних функцій він дозволяє реалізувати код рідними мовами ОС (Swift і Objective-C для iOS, а також Java і Kotlin для Android). [1]

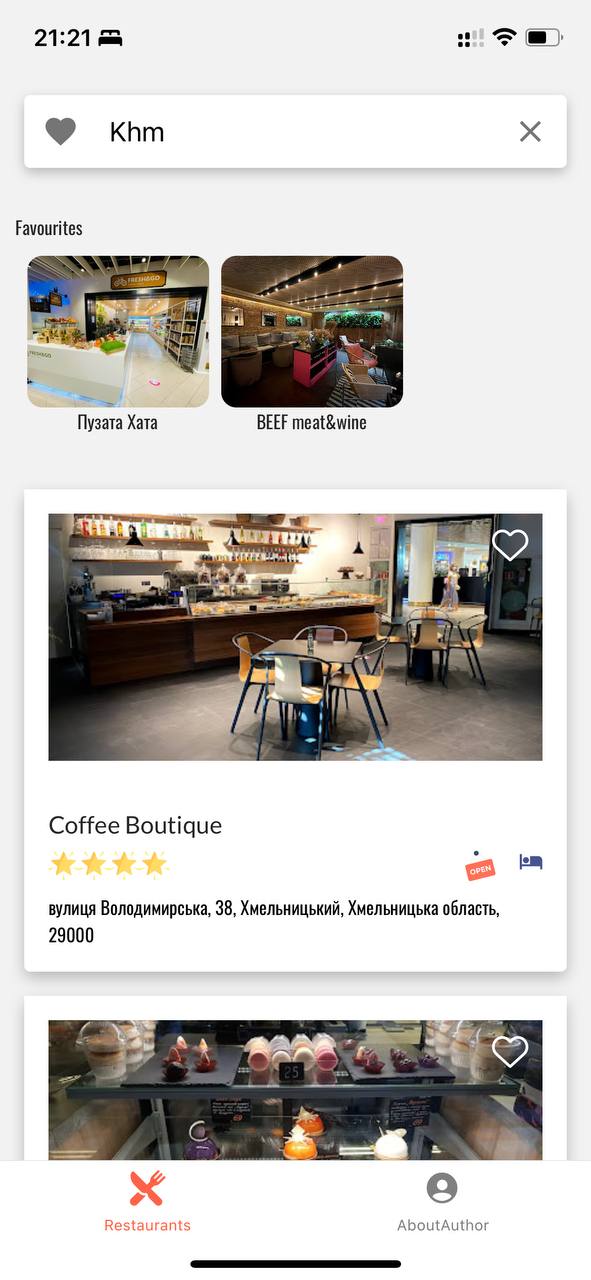
Також використовувались допоміжні бібліотеки для зображення даних у вигляді таблиці (react-native-paper), графіку (react-native-chart-kit) та збереження даних у файл (react-native-fs).

# ОПИС РОБОТИ З ПРОГРАМОЮ ТА ЗНІМКИ ЕКРАНУ

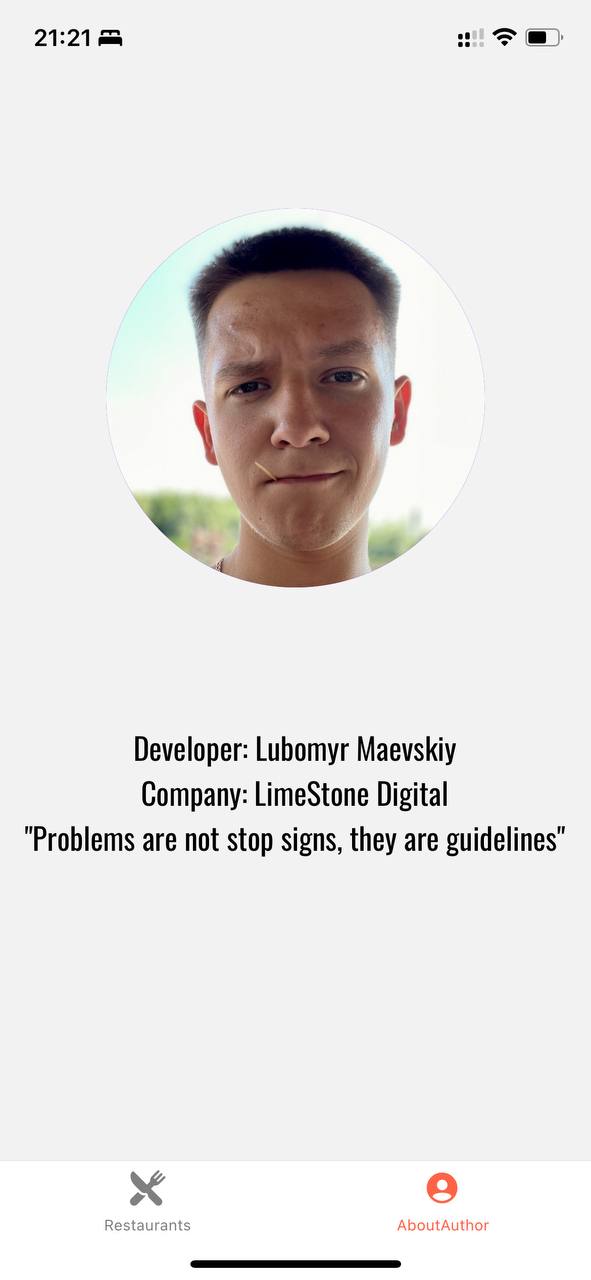
1. Початковий екран – пошук ресторанів за гео-позиціює. Навігаційна панель знаходиться у низу екрану, що є звичним для iOS.



1. Якщо нажати на кнопку “Сердечко” на ресторані, то обраний ресторан додасться до улюблених. Якщо нажати на кнопку “Сердечко” напроти пошуку, то відкриється панель із улюбленими ресторанами.



1. Екран з даними про виконавця.



# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. What is React Native And When to Use It For Your App? (Updated). *elpassion.com*. URL: <https://www.elpassion.com/blog/what-is-react-native-and-when-to-use-it> (дата звернення: 22.10.2022).
2. How to Use the Midpoint Rule in AP® Calculus. *albert.io*. URL: <https://www.albert.io/blog/how-to-use-the-midpoint-rule-in-ap-calculus/> (дата звернення: 20.10.2022).
3. Trapezoidal Rule. *byjus.com*. URL: <https://byjus.com/maths/trapezoidal-rule/> (дата звернення: 20.10.2022).
4. Simpson's Rule. *byjus.com*. URL: <https://byjus.com/maths/simpsons-rule/> (дата звернення: 20.10.2022).

# ДОДАТКИ

## ДОДАТОК А

Фрагмент Коду, Головна сторінка

import { StatusBar as ExpoStatusBar } from "expo-status-bar";

import React from "react";

import { ThemeProvider } from "styled-components/native";

import {

  useFonts as useOswald,

  Oswald\_400Regular,

} from "@expo-google-fonts/oswald";

import { useFonts as useLato, Lato\_400Regular } from "@expo-google-fonts/lato";

import { theme } from "./src/infrastructure/theme";

import { Navigation } from "./src/infrastructure/navigation";

import { RestaurantsContextProvider } from "./src/services/restaurants/restaurants.context";

import { LocationContextProvider } from "./src/services/location/location.context";

import { FavouritesContextProvider } from "./src/services/favourites/favourites.context";

export default function App() {

  const [oswaldLoaded] = useOswald({

    Oswald\_400Regular,

  });

  const [latoLoaded] = useLato({

    Lato\_400Regular,

  });

  if (!oswaldLoaded || !latoLoaded) {

    return null;

  }

  return (

    <>

      <ThemeProvider theme={theme}>

        <FavouritesContextProvider>

          <LocationContextProvider>

            <RestaurantsContextProvider>

              <Navigation />

            </RestaurantsContextProvider>

          </LocationContextProvider>

        </FavouritesContextProvider>

      </ThemeProvider>

      <ExpoStatusBar style="dark" />

    </>

  );

}

## ДОДАТОК Б

Фрагмент Коду, Пошук Ресторанів

import React, { useState, useContext, createContext, useEffect } from "react";

import {

  restaurantsRequest,

  restaurantsTransform,

} from "./restaurants.service";

import { LocationContext } from "../location/location.context";

export const RestaurantsContext = createContext();

export const RestaurantsContextProvider = ({ children }) => {

  const [restaurants, setRestaurants] = useState([]);

  const [isLoading, setIsLoading] = useState(false);

  const [error, setError] = useState(null);

  const { location } = useContext(LocationContext);

  const retrieveRestaurants = (loc) => {

    setIsLoading(true);

    setRestaurants([]);

    setTimeout(() => {

      restaurantsRequest(loc)

        .then(restaurantsTransform)

        .then((results) => {

          setIsLoading(false);

          setRestaurants(results);

        })

        .catch((err) => {

          setIsLoading(false);

          setError(err);

        });

    }, 2000);

  };

  useEffect(() => {

    if (location) {

      const locationString = `${location.lat},${location.lng}`;

      retrieveRestaurants(locationString);

    }

  }, [location]);

  return (

    <RestaurantsContext.Provider

      value={{

        restaurants,

        isLoading,

        error,

      }}

    >

      {children}

    </RestaurantsContext.Provider>

  );

};

## ДОДАТОК В

Фрагмент Коду, Відображення Ресторанів

import React, { useContext, useState } from "react";

import { FlatList, TouchableOpacity } from "react-native";

import styled from "styled-components/native";

import { ActivityIndicator, Colors } from "react-native-paper";

import { SafeArea } from "../../../components/utility/safe-area.component";

import { Spacer } from "../../../components/spacer/spacer.component";

import { FavouritesBar } from "../../../components/favourites/favourites-bar.component";

import { RestaurantsContext } from "../../../services/restaurants/restaurants.context";

import { FavouritesContext } from "../../../services/favourites/favourites.context";

import { Search } from "../components/search.component";

import { RestaurantInfoCard } from "../components/restaurant-info-card.component";

const RestaurantList = styled(FlatList).attrs({

  contentContainerStyle: {

    padding: 16,

  },

})``;

const Loading = styled(ActivityIndicator)`

  margin-left: -25px;

`;

const LoadingContainer = styled.View`

  position: absolute;

  top: 50%;

  left: 50%;

`;

export const RestaurantsScreen = ({ navigation }) => {

  const { isLoading, restaurants } = useContext(RestaurantsContext);

  const { favourites } = useContext(FavouritesContext);

  const [isToggled, setIsToggled] = useState(false);

  return (

    <SafeArea>

      {isLoading && (

        <LoadingContainer>

          <Loading size={50} animating={true} color={Colors.blue300} />

        </LoadingContainer>

      )}

      <Search

        isFavouritesToggled={isToggled}

        onFavouritesToggle={() => setIsToggled(!isToggled)}

      />

      {isToggled && (

        <FavouritesBar

          favourites={favourites}

          onNavigate={navigation.navigate}

        />

      )}

      <RestaurantList

        data={restaurants}

        renderItem={({ item }) => {

          return (

            <TouchableOpacity

              onPress={() =>

                navigation.navigate("RestaurantDetail", {

                  restaurant: item,

                })

              }

            >

              <Spacer position="bottom" size="large">

                <RestaurantInfoCard restaurant={item} />

              </Spacer>

            </TouchableOpacity>

          );

        }}

        keyExtractor={(item) => item.name}

      />

    </SafeArea>

  );

};

## ДОДАТОК Г

Фрагмент Коду, Збереження Улюблених Ресторанів

import React, { createContext, useState, useEffect } from "react";

import AsyncStorage from "@react-native-async-storage/async-storage";

export const FavouritesContext = createContext();

export const FavouritesContextProvider = ({ children }) => {

  const [favourites, setFavourites] = useState([]);

  const saveFavourites = async (value) => {

    try {

      const jsonValue = JSON.stringify(value);

      await AsyncStorage.setItem("@favourites", jsonValue);

    } catch (e) {

      console.log("error storing", e);

    }

  };

  const loadFavourites = async () => {

    try {

      const value = await AsyncStorage.getItem("@favourites");

      if (value !== null) {

        setFavourites(JSON.parse(value));

      }

    } catch (e) {

      console.log("error loading", e);

    }

  };

  const add = (restaurant) => {

    setFavourites([...favourites, restaurant]);

  };

  const remove = (restaurant) => {

    const newFavourites = favourites.filter(

      (x) => x.placeId !== restaurant.placeId

    );

    setFavourites(newFavourites);

  };

  useEffect(() => {

    loadFavourites();

  }, []);

  useEffect(() => {

    saveFavourites(favourites);

  }, [favourites]);

  return (

    <FavouritesContext.Provider

      value={{

        favourites,

        addToFavourites: add,

        removeFromFavourites: remove,

      }}

    >

      {children}

    </FavouritesContext.Provider>

  );

};