



## **SE4010 – Currents Trends in Software Engineering**

**2021.06.15**

### **Flutter Assignment**

**Batch: SE Weekend (4<sup>th</sup> year 1<sup>st</sup> semester)**

**Student ID: IT17172660**

**Jayasundara P.R.S.S**

**Group ID: 2021-REG-WE-37**

### **Group Details**

<b>SLIIT ID</b>	<b>Student Name</b>
<b>IT17172660</b>	<b>Jayasundara P.R.S.S</b>
<b>Other Members</b>	
<b>IT18001358</b>	<b>Shafkhan M.T.M</b>
<b>IT18096880</b>	<b>Hettiarachchi C.R</b>
<b>IT18087666</b>	<b>H.S Rathnasinghe</b>

## Table of Contents

Introduction .....	3
User Interfaces.....	4
1.    Splash Screen.....	4
2.    Time Zone (Location) List View .....	5
Appendix .....	6
1.    References .....	6
2.    Splash Screen Source Code .....	6
3.    Time Zone (Location) List View Source Code.....	7

## Introduction

Our group (2021-REG-WE-37) was assigned to implement a Time Zone Clock mobile application as the flutter project topic for CTSE module. As a clock app the basic functionalities include viewing the current time of the user's current time zone, and the ability to get the current time of the preferred time zone by the user. In addition to these features we developed an analog clock by using flutter animations also we added an extra functionality of adding editing and removing favorite time zones by the user, so they will be able to access the most important time zones without a hassle.

This mobile app includes Splash Screen, Digital clock view, Analog clock view to display Local and World clocks, Time Zone(location) list view and Favorite Location (Time zone) with delete and edit favorite time zone functionalities.

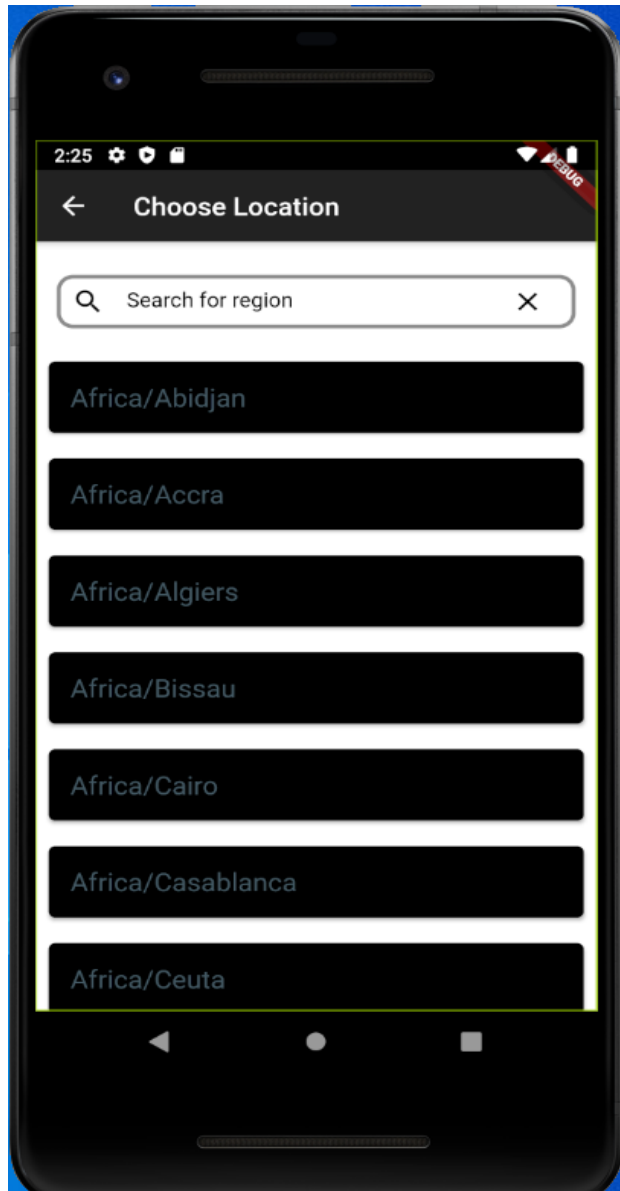
My contributions for this assignment are Splash Screen and Time Zone (Location) List View. To implement Splash screen I used Flutter native splash package, and to accommodate Time zone list I have used a list view to get all the Time zones around the world. Time Zone List View used to select Time Zone of the world as well as it includes search function to search Time Zone easily.

## User Interfaces

### 1. Splash Screen



## 2. Time Zone (Location) List View



## Appendix

### 1. References

- <https://flutter.dev/docs>
- <https://pro.academind.com/p/learn-flutter-dart-to-build-ios-android-apps-2020>
- [https://pub.dev/packages/shared\\_preferences](https://pub.dev/packages/shared_preferences)
- <https://www.youtube.com/watch?v=MPc-K8oEbdE&t=119s>

### 2. Splash Screen Source Code

```
#Splash Screen package for ISO and Android
flutter_native_splash: ^1.2.0
```

```
#For Style Flash Screen
flutter_native_splash:
  color: "#000000"
  image: assets/clock2.png
  android: true
  ios: true
```

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
  <item>
    <bitmap android:gravity="fill" android:src="@drawable/background"/>
  </item>
  <item>
    <bitmap android:gravity="center" android:src="@drawable/splash"/>
  </item>
</layer-list>
```

### 3. Time Zone (Location) List View Source Code

```
//import packages
import 'package:flutter/material.dart';
import 'package:Clock_app/consts/consts.dart';
import 'modules/search_widget.dart';

class LocationList extends StatefulWidget {
  final String selectedLocation;

  LocationList({this.selectedLocation});

  @override
  _LocationListState createState() => _LocationListState();
}

class _LocationListState extends State<LocationList> {
  var _locations;

  @override
  void initState() {
    super.initState();
    _locations = TimeLocation;
  }

  @override
  Widget build(BuildContext context) {
    return SafeArea(
      child: Scaffold(
        appBar: AppBar(
          title: Text('Choose Location'),
        ),
        backgroundColor: Color(white),
        body: Padding(
          padding: const EdgeInsets.only(top: 10.0),
          child: Column(
            children: [
              buildSearch(),
              Expanded(
                child: Center(
                  child: getListViewLocation(),
                ),
              ),
            ],
          ),
        ),
      ),
    );
  }
}
```

```

        ),
    ),
),
);
}

//build a search widget
Widget buildSearch() => SearchWidget(
    text: 'query',
    hintText: 'Search for region',
    onChanged: onSearch,
);

//get location list view
Widget getListViewLocation() {
    var listView = ListView.builder(
        itemBuilder: (context, index) {
            return Container(
                padding: EdgeInsets.all(10.0),
                child: Material(
                    elevation: 2.0,
                    borderRadius: BorderRadius.circular(5.0),
                    color: Colors.black,
                    child: Center(
                        child: ListTile(
                            title: Text(
                                _locations[index],
                                style: widget.selectedLocation == _locations[index]
                                    ? kSelectedTextStyle
                                    : TopRowTextStyle,
                            ),
                            onTap: () {
                                Navigator.pop(context, _locations[index]);
                            },
                        ),
                    ),
                )),
        itemCount: _locations.length,
    );
    return listView;
}

//search function
onSearch(String search) {
    print(search);
    final locationsArray = TimeLocation.where((location) {

```



```

        final locationLower = location.toLowerCase();

        return locationLower.contains(search);
    }).toList();

    setState(() {
        this._locations = locationsArray;
    });
}
}

```

```

//import packages
import 'package:flutter/material.dart';

//implement search widget
class SearchWidget extends StatefulWidget {
    final String text;
    final ValueChanged<String> onChanged;
    final String hintText;

    const SearchWidget({
        Key key,
        this.text,
        this.onChanged,
        this.hintText,
    }) : super(key: key);

    @override
    _SearchWidgetState createState() => _SearchWidgetState();
}

class _SearchWidgetState extends State<SearchWidget> {
    final controller = TextEditingController();

    @override
    Widget build(BuildContext context) {
        final styleActive = TextStyle(color: Colors.black);
        final styleHint = TextStyle(color: Colors.black54);
        final style = widget.text.isEmpty ? styleHint : styleActive;

        return Container(
            height: 42,

```

```

margin: const EdgeInsets.fromLTRB(16, 16, 16, 16),
decoration: BoxDecoration(
  borderRadius: BorderRadius.circular(12),
  color: Colors.white,
  border: Border.all(color: Colors.black45, width: 3),
),
padding: const EdgeInsets.symmetric(horizontal: 8),
child: TextField(
  controller: controller,
  decoration: InputDecoration(
    icon: Icon(Icons.search, color: style.color),
    suffixIcon: widget.text.isNotEmpty
      ? GestureDetector(
        child: Icon(Icons.close, color: style.color),
        onTap: () {
          controller.clear();
          widget.onChange('');
          FocusScope.of(context).requestFocus(FocusNode());
        },
      )
      : null,
    hintText: widget.hintText,
    hintStyle: style,
    border: InputBorder.none,
  ),
  style: style,
  onChanged: widget.onChange,
),
);
}
}

```

```

const int purple = 0xff3c1361;
const int red = 0xffff2525;
const int black = 0x000000;
const int brightYellow = 0xfffffe03d;
const int midYellow = 0xfffffd03d;
const int yellow = 0xfffffc03d;
const int silver = 0xffff2f5fc;
const int darkSilver = 0xffe7eefb;
const int blue = 0xff90caf9;
const int darkBlue = 0xff1565c0;
const int accentBlue = 0xff0091ea;

```

```
const int bluegrey = 0xff455a65;
const int white = 0xFFFFFFFF;

const TopRowTextStyle = TextStyle(
  fontSize: 20.0,
  fontFamily: 'Varela',
  color: Color(0xff455a65),
);

const favItemsTextStyle = TextStyle(
  fontSize: 20.0,
  fontFamily: 'Varela',
  color: Color(white),
);

const kSelectedTextStyle = TextStyle(
  fontSize: 24.0,
  fontWeight: FontWeight.bold,
  fontFamily: 'Varela',
  color: Color(white),
);

const kTimeTextStyle = TextStyle(
  fontSize: 56.0,
  fontWeight: FontWeight.bold,
  fontFamily: 'Varela',
  color: Color(white),
);

const kActiveButtonGradient = LinearGradient(
  begin: Alignment.bottomRight,
  end: Alignment.topLeft,
  colors: <Color>[
    Color(0xff009688),
    Color(0xff009688),
    Color(0xff009688),
  ],
);

const kInactiveButtonGradient = LinearGradient(
  begin: Alignment.bottomRight,
  end: Alignment.topLeft,
  colors: <Color>[
    Color(0xffe7eefb),
    Color(0xffff2f5fc),
  ],
);
```

```
    ],  
  );  
  
const defaultLocation = 'Europe/London';  
  
const TimeLocation = [  
  "Africa/Abidjan",  
  "Africa/Accra",  
  "Africa/Algiers",  
  "Africa/Bissau",  
  "Africa/Cairo",  
  "Africa/Casablanca",  
  "Africa/Ceuta",  
  "Africa/El_Aaiun",  
  "Africa/Johannesburg",  
  "Africa/Juba",  
  "Africa/Khartoum",  
  "Africa/Lagos",  
  "Africa/Maputo",  
  "Africa/Monrovia",  
  "Africa/Nairobi",  
  "Africa/Ndjamena",  
  "Africa/Sao_Tome",  
  "Africa/Tripoli",  
  "Africa/Tunis",  
  "Africa/Windhoek",  
  "America/Adak",  
  "America/Anchorage",  
  "America/Araguaina",  
  "America/Argentina/Buenos_Aires",  
  "America/Argentina/Catamarca",  
  "America/Argentina/Cordoba",  
  "America/Argentina/Jujuy",  
  "America/Argentina/La_Rioja",  
  "America/Argentina/Mendoza",  
  "America/Argentina/Rio_Gallegos",  
  "America/Argentina/Salta",  
  "America/Argentina/San_Juan",  
  "America/Argentina/San_Luis",  
  "America/Argentina/Tucuman",  
  "America/Argentina/Ushuaia",  
  "America/Asuncion",  
  "America/Atikokan",  
  "America/Bahia",  
  "America/Bahia_Banderas",
```

"America/Barbados",  
"America/Belem",  
"America/Belize",  
"America/Blanc-Sablon",  
"America/Boa\_Vista",  
"America/Bogota",  
"America/Boise",  
"America/Cambridge\_Bay",  
"America/Campo\_Grande",  
"America/Cancun",  
"America/Caracas",  
"America/Cayenne",  
"America/Chicago",  
"America/Chihuahua",  
"America/Costa\_Rica",  
"America/Creston",  
"America/Cuiaba",  
"America/Curacao",  
"America/Danmarkshavn",  
"America/Dawson",  
"America/Dawson\_Creek",  
"America/Denver",  
"America/Detroit",  
"America/Edmonton",  
"America/Eirunepe",  
"America/El\_Salvador",  
"America/Fort\_Nelson",  
"America/Fortaleza",  
"America/Glace\_Bay",  
"America/Goose\_Bay",  
"America/Grand\_Turk",  
"America/Guatemala",  
"America/Guayaquil",  
"America/Guyana",  
"America/Halifax",  
"America/Havana",  
"America/Hermosillo",  
"America/Indiana/Indianapolis",  
"America/Indiana/Knox",  
"America/Indiana/Marengo",  
"America/Indiana/Petersburg",  
"America/Indiana/Tell\_City",  
"America/Indiana/Vevay",  
"America/Indiana/Vincennes",  
"America/Indiana/Winamac",

"America/Inuvik",  
"America/Iqaluit",  
"America/Jamaica",  
"America/Juneau",  
"America/Kentucky/Louisville",  
"America/Kentucky/Monticello",  
"America/La\_Paz",  
"America/Lima",  
"America/Los\_Angeles",  
"America/Maceio",  
"America/Managua",  
"America/Manaus",  
"America/Martinique",  
"America/Matamoros",  
"America/Mazatlan",  
"America/Menominee",  
"America/Merida",  
"America/Metlakatla",  
"America/Mexico\_City",  
"America/Miquelon",  
"America/Moncton",  
"America/Monterrey",  
"America/Montevideo",  
"America/Nassau",  
"America/New\_York",  
"America/Nipigon",  
"America/Nome",  
"America/Noronha",  
"America/North\_Dakota/Beulah",  
"America/North\_Dakota/Center",  
"America/North\_Dakota/New\_Salem",  
"America/Nuuk",  
"America/Ojinaga",  
"America/Panama",  
"America/Pangnirtung",  
"America/Paramaribo",  
"America/Phoenix",  
"America/Port-au-Prince",  
"America/Port\_of\_Spain",  
"America/Porto\_Velho",  
"America/Puerto\_Rico",  
"America/Punta\_Arenas",  
"America/Rainy\_River",  
"America/Rankin\_Inlet",  
"America/Recife",

"America/Regina",  
"America/Resolute",  
"America/Rio\_Branco",  
"America/Santarem",  
"America/Santiago",  
"America/Santo\_Domingo",  
"America/Sao\_Paulo",  
"America/Scoresbysund",  
"America/Sitka",  
"America/St\_Johns",  
"America/Swift\_Current",  
"America/Tegucigalpa",  
"America/Thule",  
"America/Thunder\_Bay",  
"America/Tijuana",  
"America/Toronto",  
"America/Vancouver",  
"America/Whitehorse",  
"America/Winnipeg",  
"America/Yakutat",  
"America/Yellowknife",  
"Antarctica/Casey",  
"Antarctica/Davis",  
"Antarctica/DumontDUrville",  
"Antarctica/Macquarie",  
"Antarctica/Mawson",  
"Antarctica/Palmer",  
"Antarctica/Rothera",  
"Antarctica/Syowa",  
"Antarctica/Troll",  
"Antarctica/Vostok",  
"Asia/Almaty",  
"Asia/Amman",  
"Asia/Anadyr",  
"Asia/Aqtau",  
"Asia/Aqtobe",  
"Asia/Ashgabat",  
"Asia/Atyrau",  
"Asia/Baghdad",  
"Asia/Baku",  
"Asia/Bangkok",  
"Asia/Barnaul",  
"Asia/Beirut",  
"Asia/Bishkek",  
"Asia/Brunei",

"Asia/Chita",  
"Asia/Choibalsan",  
"Asia/Colombo",  
"Asia/Damascus",  
"Asia/Dhaka",  
"Asia/Dili",  
"Asia/Dubai",  
"Asia/Dushanbe",  
"Asia/Famagusta",  
"Asia/Gaza",  
"Asia/Hebron",  
"Asia/Ho\_Chi\_Minh",  
"Asia/Hong\_Kong",  
"Asia/Hovd",  
"Asia/Irkutsk",  
"Asia/Jakarta",  
"Asia/Jayapura",  
"Asia/Jerusalem",  
"Asia/Kabul",  
"Asia/Kamchatka",  
"Asia/Karachi",  
"Asia/Kathmandu",  
"Asia/Khandyga",  
"Asia/Kolkata",  
"Asia/Krasnoyarsk",  
"Asia/Kuala\_Lumpur",  
"Asia/Kuching",  
"Asia/Macau",  
"Asia/Magadan",  
"Asia/Makassar",  
"Asia/Manila",  
"Asia/Nicosia",  
"Asia/Novokuznetsk",  
"Asia/Novosibirsk",  
"Asia/Omsk",  
"Asia/Oral",  
"Asia/Pontianak",  
"Asia/Pyongyang",  
"Asia/Qatar",  
"Asia/Qostanay",  
"Asia/Qyzylorda",  
"Asia/Riyadh",  
"Asia/Sakhalin",  
"Asia/Samarkand",  
"Asia/Seoul",



"Asia/Shanghai",  
"Asia/Singapore",  
"Asia/Srednekolymsk",  
"Asia/Taipei",  
"Asia/Tashkent",  
"Asia/Tbilisi",  
"Asia/Tehran",  
"Asia/Thimphu",  
"Asia/Tokyo",  
"Asia/Tomsk",  
"Asia/Ulaanbaatar",  
"Asia/Urumqi",  
"Asia/Ust-Nera",  
"Asia/Vladivostok",  
"Asia/Yakutsk",  
"Asia/Yangon",  
"Asia/Yekaterinburg",  
"Asia/Yerevan",  
"Atlantic/Azores",  
"Atlantic/Bermuda",  
"Atlantic/Canary",  
"Atlantic/Cape\_Verde",  
"Atlantic/Faroe",  
"Atlantic/Madeira",  
"Atlantic/Reykjavik",  
"Atlantic/South\_Georgia",  
"Atlantic/Stanley",  
"Australia/Adelaide",  
"Australia/Brisbane",  
"Australia/Broken\_Hill",  
"Australia/Currie",  
"Australia/Darwin",  
"Australia/Eucla",  
"Australia/Hobart",  
"Australia/Lindeman",  
"Australia/Lord\_Howe",  
"Australia/Melbourne",  
"Australia/Perth",  
"Australia/Sydney",  
"CET",  
"CST6CDT",  
"EET",  
"EST",  
"EST5EDT",  
"Etc/GMT",

```
"Etc/GMT+1",
"Etc/GMT+10",
"Etc/GMT+11",
"Etc/GMT+12",
"Etc/GMT+2",
"Etc/GMT+3",
"Etc/GMT+4",
"Etc/GMT+5",
"Etc/GMT+6",
"Etc/GMT+7",
"Etc/GMT+8",
"Etc/GMT+9",
"Etc/GMT-1",
"Etc/GMT-10",
"Etc/GMT-11",
"Etc/GMT-12",
"Etc/GMT-13",
"Etc/GMT-14",
"Etc/GMT-2",
"Etc/GMT-3",
"Etc/GMT-4",
"Etc/GMT-5",
"Etc/GMT-6",
"Etc/GMT-7",
"Etc/GMT-8",
"Etc/GMT-9",
"Etc/UTC",
"Europe/Amsterdam",
"Europe/Andorra",
"Europe/Astrakhan",
"Europe/Athens",
"Europe/Belgrade",
"Europe/Berlin",
"Europe/Brussels",
"Europe/Bucharest",
"Europe/Budapest",
"Europe/Chisinau",
"Europe/Copenhagen",
"Europe/Dublin",
"Europe/Gibraltar",
"Europe/Helsinki",
"Europe/Istanbul",
"Europe/Kaliningrad",
"Europe/Kiev",
"Europe/Kirov",
```

"Europe/Lisbon",  
"Europe/London",  
"Europe/Luxembourg",  
"Europe/Madrid",  
"Europe/Malta",  
"Europe/Minsk",  
"Europe/Monaco",  
"Europe/Moscow",  
"Europe/Oslo",  
"Europe/Paris",  
"Europe/Prague",  
"Europe/Riga",  
"Europe/Rome",  
"Europe/Samara",  
"Europe/Saratov",  
"Europe/Simferopol",  
"Europe/Sofia",  
"Europe/Stockholm",  
"Europe/Tallinn",  
"Europe/Tirane",  
"Europe/Ulyanovsk",  
"Europe/Uzhgorod",  
"Europe/Vienna",  
"Europe/Vilnius",  
"Europe/Volgograd",  
"Europe/Warsaw",  
"Europe/Zaporozhye",  
"Europe/Zurich",  
"HST",  
"Indian/Chagos",  
"Indian/Christmas",  
"Indian/Cocos",  
"Indian/Kerguelen",  
"Indian/Mahe",  
"Indian/Maldives",  
"Indian/Mauritius",  
"Indian/Reunion",  
"MET",  
"MST",  
"MST7MDT",  
"PST8PDT",  
"Pacific/Apia",  
"Pacific/Auckland",  
"Pacific/Bougainville",  
"Pacific/Chatham",

```
"Pacific/Chuuk",  
"Pacific/Easter",  
"Pacific/Efate",  
"Pacific/Enderbury",  
"Pacific/Fakaofo",  
"Pacific/Fiji",  
"Pacific/Funafuti",  
"Pacific/Galapagos",  
"Pacific/Gambier",  
"Pacific/Guadalcanal",  
"Pacific/Guam",  
"Pacific/Honolulu",  
"Pacific/Kiritimati",  
"Pacific/Kosrae",  
"Pacific/Kwajalein",  
"Pacific/Majuro",  
"Pacific/Marquesas",  
"Pacific/Nauru",  
"Pacific/Niue",  
"Pacific/Norfolk",  
"Pacific/Noumea",  
"Pacific/Pago_Pago",  
"Pacific/Palau",  
"Pacific/Pitcairn",  
"Pacific/Pohnpei",  
"Pacific/Port_Moresby",  
"Pacific/Rarotonga",  
"Pacific/Tahiti",  
"Pacific/Tarawa",  
"Pacific/Tongatapu",  
"Pacific/Wake",  
"Pacific/Wallis",  
"WET"
```

```
];
```