

Experiment 05: Write a program to implement date and time picker

Learning Objective: Student should be able to write a program to implement date and time picker in Android Studio.

Tools: Android Studio

Theory:

Android Studio:

Android Studio is the official integrated development environment (IDE) for Android application development. Every project in Android Studio has one or more modalities with source code and resource files. These modalities include Android app modules, Library modules, and Google App Engine modules. It is an open source software platform and operating system for mobile devices which is based on the Linux kernel. It allows writing managed code in the Java language and Kotlin language. Android apps are built as a combination of components that can be invoked individually. For example, an activity is a type of app component that provides a user interface (UI). The "main" activity starts when the user taps the application's icon. One can also direct the user to an activity from elsewhere, such as from a notification or even from a different app. Other components, such as broadcast receivers and services, allow the application to perform background tasks without a UI.

Data and Time Picker:

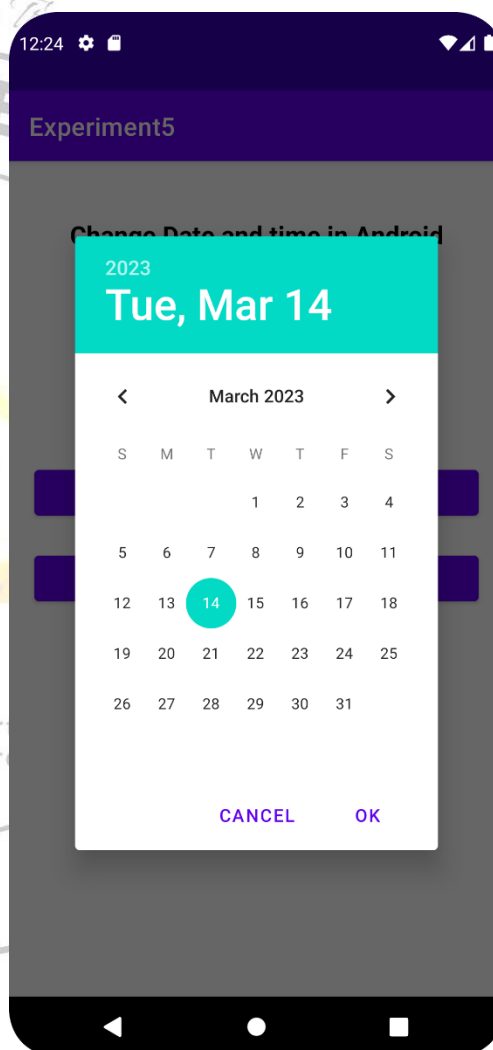
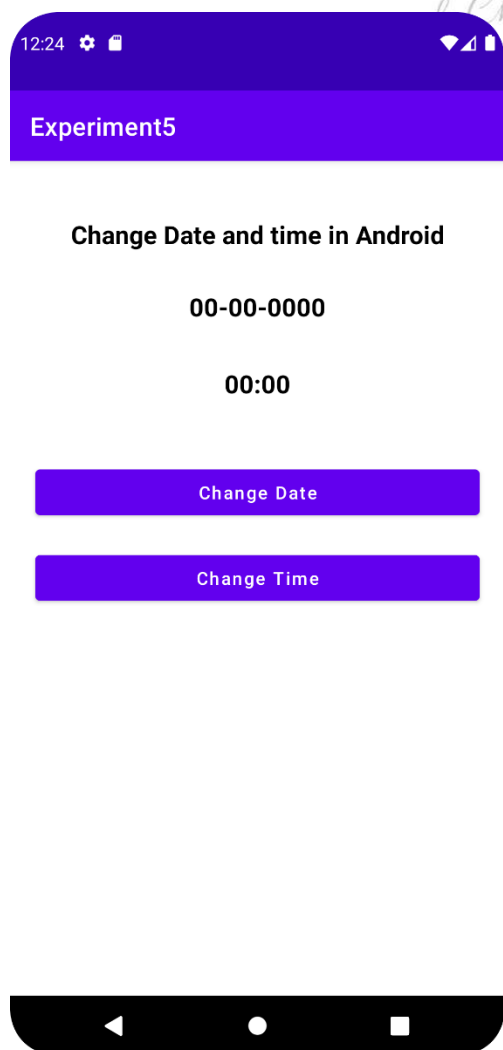
Android Date Picker allows coder to select the date consisting of day, month and year in the custom user interface. For this functionality android provides DatePicker and DatePickerDialog components.

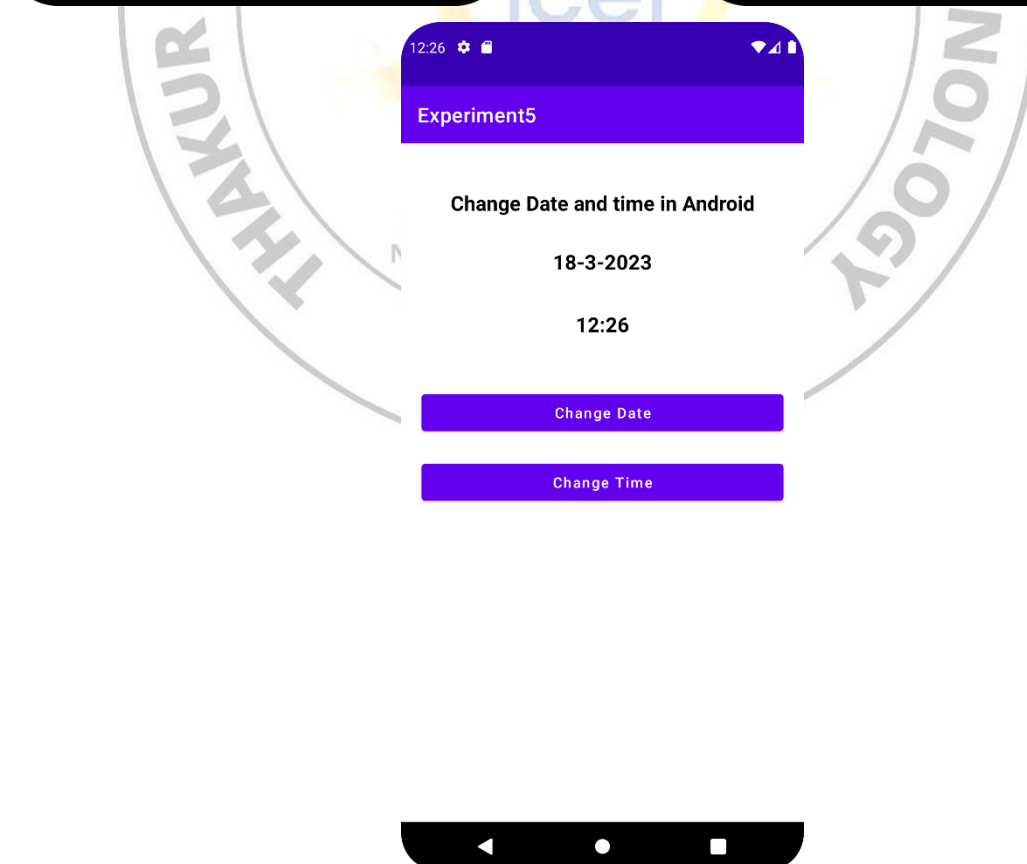
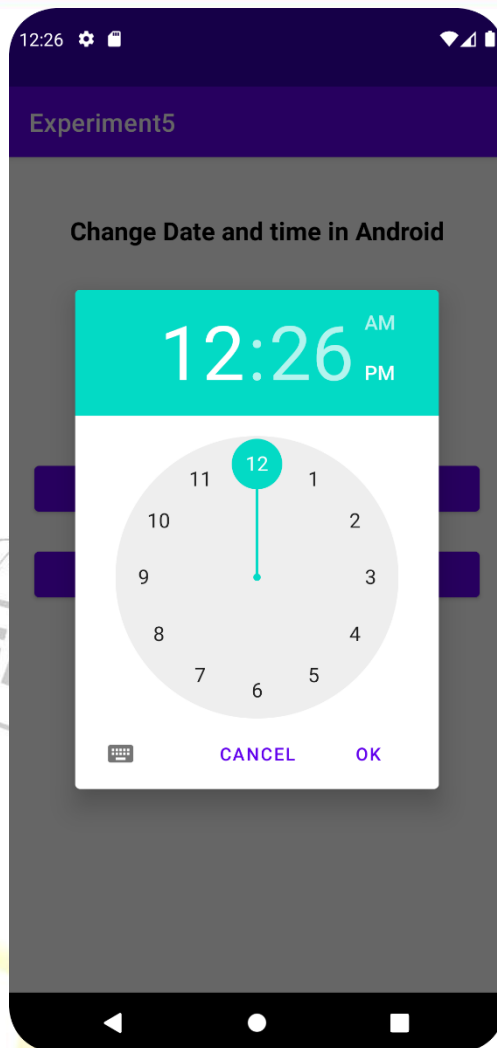
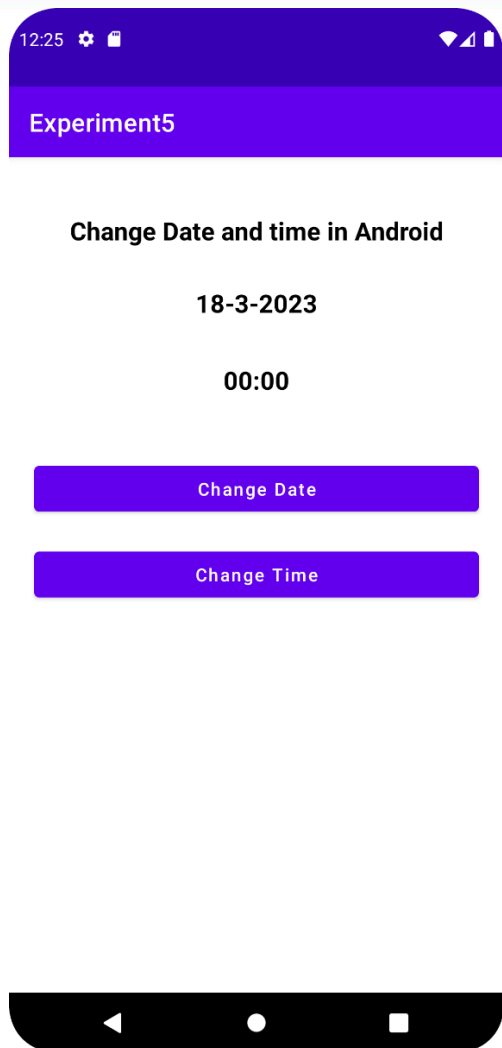
The DatePickerDialog class consists of a 5 argument constructor with the parameters listed below.

- Context: It requires the application context
 - Callback Function: `onDateSet()` is invoked when the user sets the date with the following parameters:
 - `int year` : It will be store the current selected year from the dialog
 - `int monthOfYear` : It will be store the current selected month from the dialog
 - `int dayOfMonth` : It will be store the current selected day from the dialog
 - `int mYear` : It shows the the current year that's visible when the dialog pops up
 - `int mMonth` : It shows the the current month that's visible when the dialog pops up
 - `int mDay` : It shows the the current day that's visible when the dialog pops up

- The TimePickerDialog class consists of a 5 argument constructor with the parameters listed below.
 - Context: It requires the application context
 - Callback Function: onTimeSet() is invoked when the user sets the time with the following parameters:
 - int hourOfDay : It will be store the current selected hour of the day from the dialog
 - int minute : It will be store the current selected minute from the dialog
 - int mHours : It shows the the current Hour that's visible when the dialog pops up
 - int mMinute : It shows the the current minute that's visible when the dialog pops up
 - boolean false : If its set to false it will show the time in 24 hour format else not

Implementation:





MainActivity.java:

```
package com.example.experiment5;
```

```
import android.app.DatePickerDialog;
```

```
import android.app.TimePickerDialog;
```

```
import android.os.Bundle;
```

```
import android.view.View;
```

```
import android.widget.Button;
```

```
import android.widget.DatePicker;
```

```
import android.widget.TimePicker;
```

```
import android.widget.TextView;
```

```
import
```

```
androidx.appcompat.app.AppCompatActivity;
```

```
vity;
```

```
import java.util.Calendar;
```

```
public class MainActivity extends
```

```
AppCompatActivity {
```

```
    // on below line we are creating
    variables.
```

```
    private Button pickDateBtn;
```

```
    private TextView selectedDateTV;
```

```
    private Button pickTimeBtn;
```

```
    private TextView selectedTimeTV;
```

```
    @Override
```

```
    protected void onCreate(Bundle
    savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        // on below line we are initializing
        our variables.
```

```
        pickDateBtn =
```

```
        findViewById(R.id.idBtnPickDate);
```

```
        selectedDateTV =
```

```
        findViewById(R.id.idTVSelectedDate);
```

```
        pickTimeBtn =
```

```
        findViewById(R.id.idBtnPickTime);
```

```
        selectedTimeTV =
```

```
        findViewById(R.id.idTVSelectedTime);
```

```
        // on below line we are adding click
        listener for our pick date button
```

```
        pickDateBtn.setOnClickListener(new
        View.OnClickListener() {
```

```
            @Override
```

```
            public void onClick(View v) {
```

```
                // on below line we are getting
```

```
                // the instance of our calendar.
```

```
                final Calendar c =
```

```
                Calendar.getInstance();
```

```
                // on below line we are getting
```

```
                // our day, month and year.
```

```
                int year =
```

```
                c.get(Calendar.YEAR);
```

```
                int month =
```

```
                c.get(Calendar.MONTH);
```

```
                int day =
```

```
                c.get(Calendar.DAY_OF_MONTH);
```

```
                // on below line we are creating
                a variable for date picker dialog.
```

```
                DatePickerDialog
```

```
                datePickerDialog = new
```

```
                DatePickerDialog(
```

```
                // on below line we are
                passing context.
```

```
                MainActivity.this,
```

```
                new
```

```
                DatePickerDialog.OnDateSetListener() {
```

```
                    @Override
```

```
                    public void
```

```
                    onDateSet(DatePicker view, int year,
```

```
                        int
```

```
                        monthOfYear, int dayOfMonth) {
```

```
                            // on below line we are
                            setting date to our text view.
```

```
                            selectedDateTV.setText(dayOfMonth + "-"
                            + (monthOfYear + 1) + "-" + year);
```

```
                    }
```

```
                },
```

```
                // on below line we are
                passing year,
```

```
                // month and day for
                selected date in our date picker.
```



```

        year, month, day);
    // at last we are calling show to
    // display our date picker dialog.
    datePickerDialog.show();
}
});

```

```

pickTimeBtn.setOnClickListener(new
View.OnClickListener() {

```

```

    @Override
    public void onClick(View v) {
        // on below line we are getting

```

the

```

        // instance of our calendar.

```

```

        final Calendar c =
Calendar.getInstance();

```

```

        // on below line we are getting
our hour, minute.

```

```

        int hour =
c.get(Calendar.HOUR_OF_DAY);

```

```

        int minute =
c.get(Calendar.MINUTE);

```

```

        // on below line we are
initializing our Time Picker Dialog

```

activity_main.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout

```

```

xmlns:android="http://schemas.android.co
m/apk/res/android"

```

```

xmlns:tools="http://schemas.android.com/
tools"

```

```

    android:id="@+id/idRLLContainer"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

```

```

<!--on below line we are creating
a text for our app-->

```

TimePickerDialog

```

timePickerDialog = new
TimePickerDialog(MainActivity.this,
new

```

```

TimePickerDialog.OnTimeSetListener() {
    @Override
    public void
onTimeSet(TimePicker view, int
hourOfDay,
int minute)

```

```

{
    // on below line we are
setting selected time
    // in our text view.

```

```

    selectedTimeTV.setText(hourOfDay + ":"
+ minute);

```

```

}
}, hour, minute, false);

```

```

    // at last we are calling show to
    // display our time picker dialog.
    timePickerDialog.show();

```

```

}
});

```

```

}
}

```

```

<!--on below line we are creating a text
view-->

```

```

<!--on below line we are creating
a button for date picker-->

```

```

<TextView
    android:id="@+id/idTVHeading"

```

```

    android:layout_width="match_parent"

```

```

    android:layout_height="wrap_content"

```

```

    android:layout_above="@id/idTVSelecte
dDate"

```

```

    android:layout_centerInParent="true"
    android:layout_marginStart="20dp"

```

```
android:layout_marginTop="20dp"
android:layout_marginEnd="20dp"
android:layout_marginBottom="0dp"
android:gravity="center"
android:padding="10dp"
android:text="Change Date and time
```

in Android"

```
android:textAlignment="center"
android:textColor="@color/black"
android:textSize="20sp"
android:textStyle="bold" />
```

<TextView

```
android:id="@+id/idTVSelectedDate"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
```

```
android:layout_above="@id/idBtnPickDate"
```

```
android:layout_centerInParent="true"
android:layout_marginStart="20dp"
android:layout_marginTop="10dp"
android:layout_marginEnd="20dp"
```

```
android:layout_marginBottom="80dp"
```

```
android:gravity="center"
```

```
android:padding="10dp"
```

```
android:text="00-00-0000"
```

```
android:textAlignment="center"
```

```
android:textColor="@color/black"
```

```
android:textSize="20sp"
```

```
android:textStyle="bold" />
```

<Button

```
android:id="@+id/idBtnPickDate"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
```

```
android:layout_above="@+id/idBtnPickTime"
```

```
android:layout_centerInParent="true"
```

```
android:layout_marginStart="20dp"
```

```
android:layout_marginTop="20dp"
```

```
android:layout_marginEnd="20dp"
```

```
android:text="Change Date"
```

```
android:textAllCaps="false" />
```

<TextView

```
android:id="@+id/idTVSelectedTime"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
```

```
android:layout_above="@id/idBtnPickDate"
```

```
android:layout_centerInParent="true"
```

```
android:layout_marginStart="20dp"
```

```
android:layout_marginTop="20dp"
```

```
android:layout_marginEnd="20dp"
```

```
android:layout_marginBottom="19dp"
```

```
android:gravity="center"
```

```
android:padding="10dp"
```

```
android:text="00:00"
```

```
android:textAlignment="center"
```

```
android:textColor="@color/black"
```

```
android:textSize="20sp"
```

```
android:textStyle="bold" />
```

<Button

```
android:id="@+id/idBtnPickTime"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
```

```
android:layout_centerInParent="true"
```

```
android:layout_marginStart="20dp"
```

```
android:layout_marginTop="20dp"
```

```
android:layout_marginEnd="20dp"
```

```
android:layout_marginBottom="20dp"
```

```
android:text="Change Time"
```

```
android:textAllCaps="false" />
```

</RelativeLayout>

Result and Discussion: We successfully implemented a program to implement date and time picker in Android Studio.

Learning Outcomes: The student should have the ability to

LO1: explain how a date and time picker can be implemented in Android Studio.

LO2: execute a simple program to implement date and time picker in Android Studio

Course Outcomes: Upon completion of the course students will be able to to implement date and time picker in Android Studio

Conclusion:

We understood in this experiment the concept of date and time picker in android studios. Furthermore the DatePickerDialog class and TimePickerDialog class along with their respective parameters were understood. We implemented program that allows user to pick their date of birth and select a particular time slot. The program was executed on the Android VirtualDevice.

For Faculty Use

Correction Parameters	Formative Assessment [40%]	Timely completion of Practical [40%]	Attendance / Learning Attitude [20%]	
Marks Obtained				

