MINISTERUL EDUCAŢIEI REPUBLICII MOLDOVA

UNIVERSITATEA TEHNICĂ A MOLDOVEI

Facultatea „Calculatoare, Informatică şi Microelectronică”

FILIERA ANGLOFONĂ

RAPORT

Lucrare de laborator nr. 2

la Programarea Aplicațiilor Mobile

A efectuat:

st. gr. FAF-151 Bîzdîga Stanislav

A verificat:

asist.univ. Sergiu Ciudin

Chișinău-2017

Laboratory Work 2

Subject: Organizer mobile application (Daily Planner)

Functional: Sunset sightseeing planner

Theme: Sunset Reminder

Purpose of work:

Create an app of type organizer on Android.

Forming and projecting has some intended limits noted further, but the design, API and framework is free to be chosen at will. The components and structure of each activity is described below.

Tasks:

UI Components

Application should contain minimum 3 basic activities, that will follow as:

1. **MainActivity** (structure/components)

- Calendar View (custom or default)
- Buttons (Add/Remove/Update)
- Search (by keywords)

2. **AddActivity**

- Data/Time controller
- Info TextBox
- Buttons
- other (based on specifics of the app)

3. **UpdateActivity** - similar to AddActivity, just already completed

Operational data from inside the app will be stored in XML/JSON files. (keywords, **XML Serialization**).

Logical/Operational Component

All events and actions of notification (audio/visual) done by organizer should be treated by a **special service**, that functionally will extract the data from XML/JSON file.

The topic and functionality of the app is free to be chosen by the student.

Introduction

The following work is of extreme importance regarding the fact, that it is the second mobile application development experience. It is supposed to form an understanding of complex, more advanced UI elements, alarm managers, events, parsing, storing and accessing data, and possibly more.
The amount of tasks seems small but each has its own special category of things to be learnt.
However, anything that's challenging, is what aids to improve the skill greatly.

Short Theory

1. Calendar view

Calendar view is a widget in android that allows for displaying and selecting dates from a configurable and pre-set range.[1] Users can select the dates by tapping/clicking on them.
Some important methods to set the calendar:

`setFirstDayOfWeek(int firstDayOfWeek)`
This method is used to set the first day of the week.

`setMaxDate(long maxDate) / setMinDate(long minDate)`
This method is used to set the maximal / minimal date supported by thisCalendarView in milliseconds since January 1, 1970 00:00:00

`setShowWeekNumber(boolean showWeekNumber)`
This method is used to display or hide the week number of CalendarView. In this method it is possible to set the value either true or false.

2. Searching

In order to search things in a string, there's a very handy function that's called **contains( )**

With this method, any string variable may return true if it contains that parameter that was inputted in the function. It works like this:

```
MyString.contains("love");
```

Task Implementation

1. Designing the **MainActivity** with Calendar View, Buttons (Add/Remove/Update) and Search Bar:

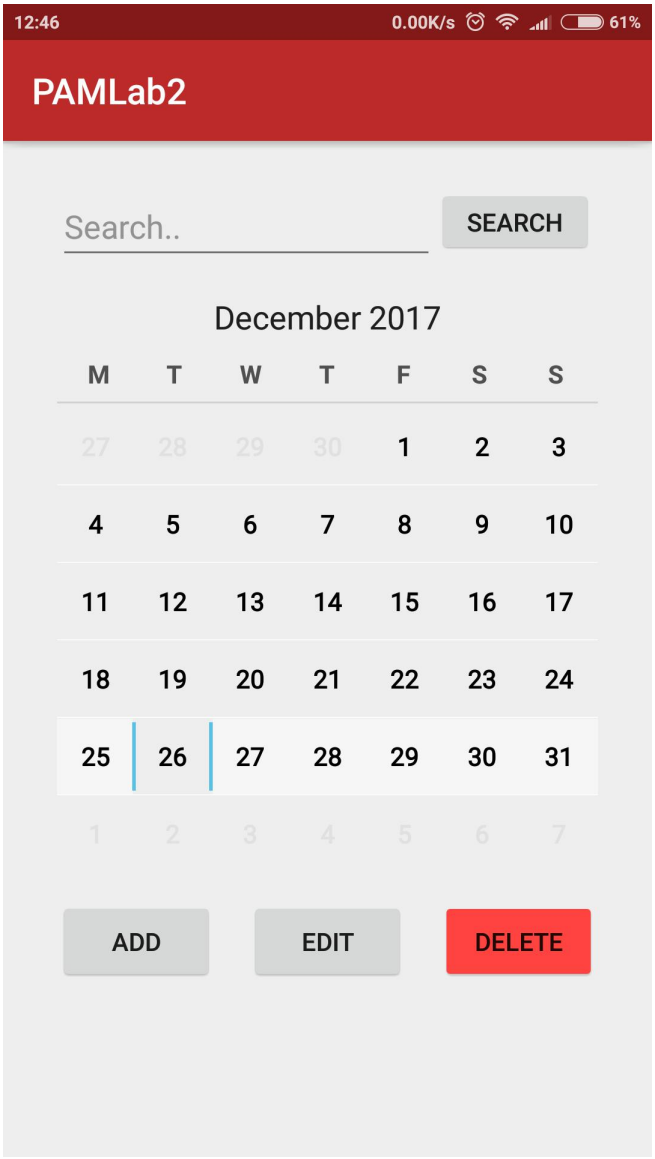

*Fig.1 The Design and Layout of **MainActivity***

The following snippet contains the declaration of **MainActivity**:

```
public class MainActivity extends AppCompatActivity {
...
```

And linking it to the layout file:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
...
```

In the layout, adding and setting the CalendarView is completed with the following snippet of code:

```
<CalendarView
    android:id="@+id/simpleCalendarView"
    android:layout_width="300dp"
    android:layout_height="330dp"
    android:firstDayOfWeek="2"
    android:maxDate="01.01.2019"
    android:minDate="01.01.2017"
    android:selectedWeekBackgroundColor="?android:attr/textColorHintInverse"
    android:showWeekNumber="false"
    android:unfocusedMonthDateColor="@color/colorLGrey"
    android:weekSeparatorLineColor="@android:color/background_light"
    android:layout_marginTop="5dp"
    android:layout_below="@+id/editTextSearch"
    android:layout_centerHorizontal="true" />
```

Then the buttons, similarly:

```
<Button
    android:id="@+id/btnEdit"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/simpleCalendarView"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="15dp"
    android:text="Edit" />

<Button
    android:id="@+id/btnDel"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/simpleCalendarView"
    android:layout_alignEnd="@+id/simpleCalendarView"
    android:backgroundTint="@color/colorAccent"
    android:layout_marginTop="15dp"
    android:text="Delete" />
```

And search bar with its button:

```
<EditText
    android:id="@+id/editTextSearch"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:ems="10"
    android:inputType="text"
    android:hint="Search.."
    android:layout_marginTop="25dp"
    android:layout_alignStart="@+id/simpleCalendarView" />

<Button
    android:id="@+id/btnSearch"
    android:layout_width="wrap_content"
    android:layout_height="40dp"
    android:text="Search"
    android:layout_marginTop="25dp"
    android:layout_toEndOf="@+id/editTextSearch" />
```

All these elements are connected to the activity during the time onCreate method is running:

```
final Context context = getApplicationContext();
final EditText textToSearch = (EditText) findViewById(R.id.editTextSearch);
final Button addButton = (Button) findViewById(R.id.btnAdd);
final Button editButton = (Button) findViewById(R.id.btnEdit);
final Button deleteButton = (Button) findViewById(R.id.btnDel);
final Button searchButton = (Button) findViewById(R.id.btnSearch);
final CalendarView simpleCalendarView = (CalendarView) findViewById(R.id.simpleCalendarView);
// get reference
final Calendar selected = Calendar.getInstance();
```

2. Designing the **AddActivity** with Data/Time controller, Info TextBox, Buttons and other:

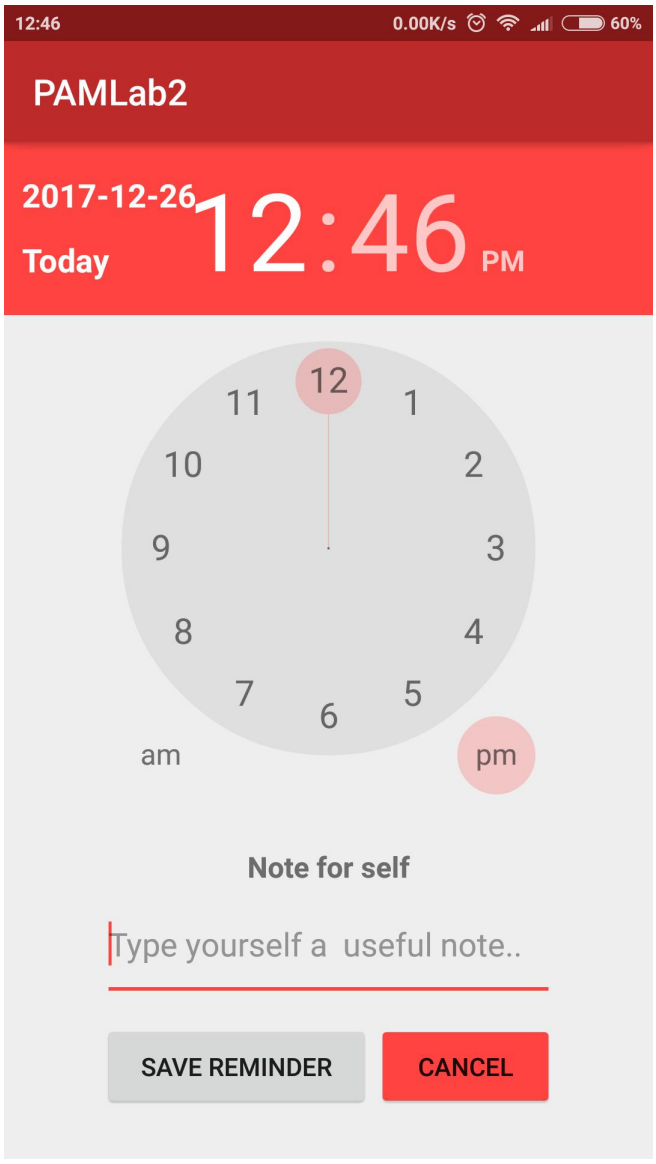

*Fig.2 The Design and Layout of **AddActivity** (also **EditActivity**)*

The Data/Time controller allows to pick the time of alarm, while the note will be useful for search and just reminding the user whatever necessary. The cancel button cancels the creation of the event, and the save button, saves it, launching the reminder at the set time.

3. Designing the **EditActivity** is implemented simply copying the AddActivity code and modifying it so that it is completing itself from the files with saved data.

This is completed as follows:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_add);
...
```

Notice that we set the same AddActivity layout, even if it is another activity - EditActivity.

Next, the references for the layout elements are set, and the date and day information is set using the extras sent previously along with the intent when the respective button was clicked.

```
    final Button saveButton = (Button) findViewById(R.id.btnSave);
    final Button cancelButton = (Button) findViewById(R.id.btnCancel);
    TextView Today = (TextView) findViewById(R.id.textView1);
    TextView weekday = (TextView) findViewById(R.id.textView2);

    day = (String) getIntent().getExtras().get("day");
    date = (String) getIntent().getExtras().get("date");
    Today.setText(date);
    weekday.setText(day);


    final TimePicker timePicker = (TimePicker) findViewById(R.id.timePicker);
    final EditText editNote = (EditText) findViewById(R.id.editNote);
```

Lastly, the data is being read from the file:

```
JSONTokener reader = new JSONTokener(FileHandler.readFromFile(getApplicationContext(), date));
```

And then written into the activity elements:

```
    try {
        JSONObject object = (JSONObject) reader.nextValue();
        String jsonnote = object.getString("note");
        editNote.setText(jsonnote);

        String jsontime[] = object.getString("time").split(":");
        timePicker.setCurrentHour(Integer.valueOf(jsontime[0]));
        timePicker.setCurrentMinute(Integer.valueOf(jsontime[1]));
    } catch (JSONException e) {
        e.printStackTrace();
    }
...
```

In case there was no file found, EditActivity acts exactly similar to AddActivity.

4. Worth mentioning, that the files are saved in JSON , being entitled as their respective dates, thus each event is separate and unique. And since the app is aimed to remind you just one thing a day, because the Sun sets just once a day, there's only one event per day that can be added and edited (overwritten).

Results and verification

Testing the alarm system:

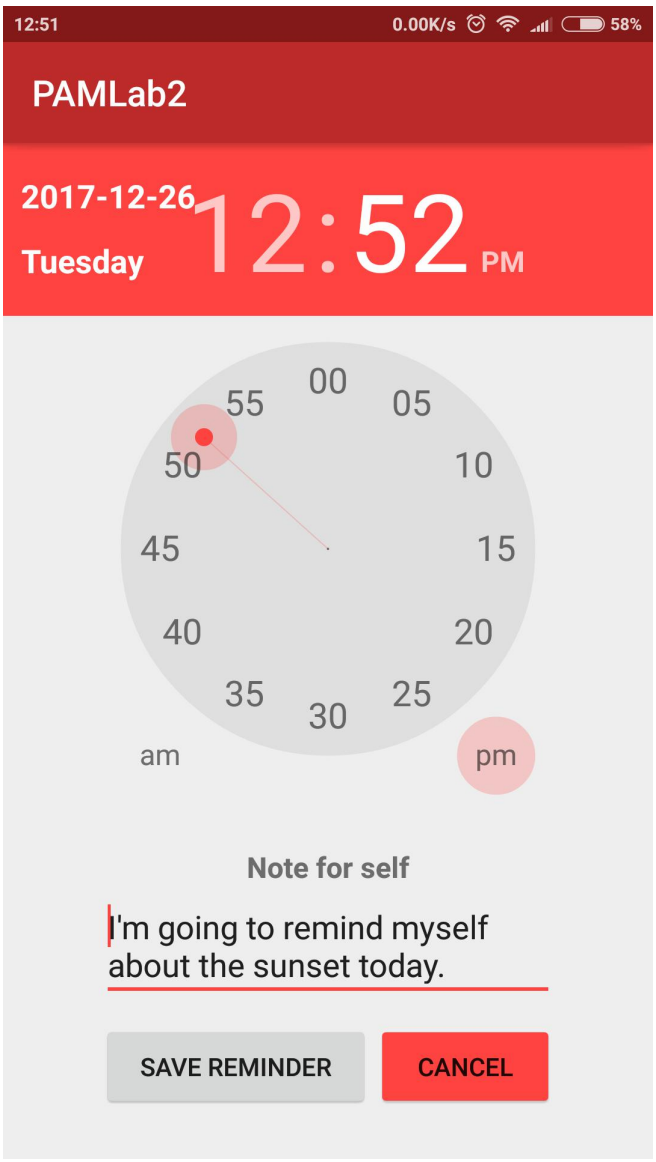

fig.3. Creating a new reminder for 12:52

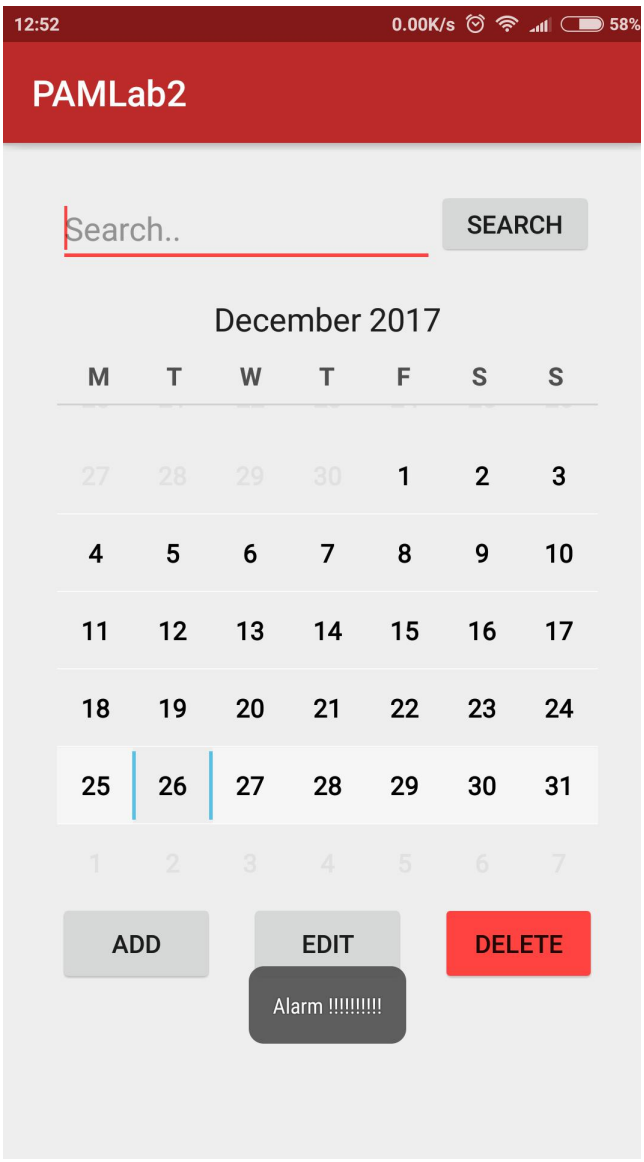

fig.4. The alarm enables at 12:52 and vibrates

Verifying the search and deletion:

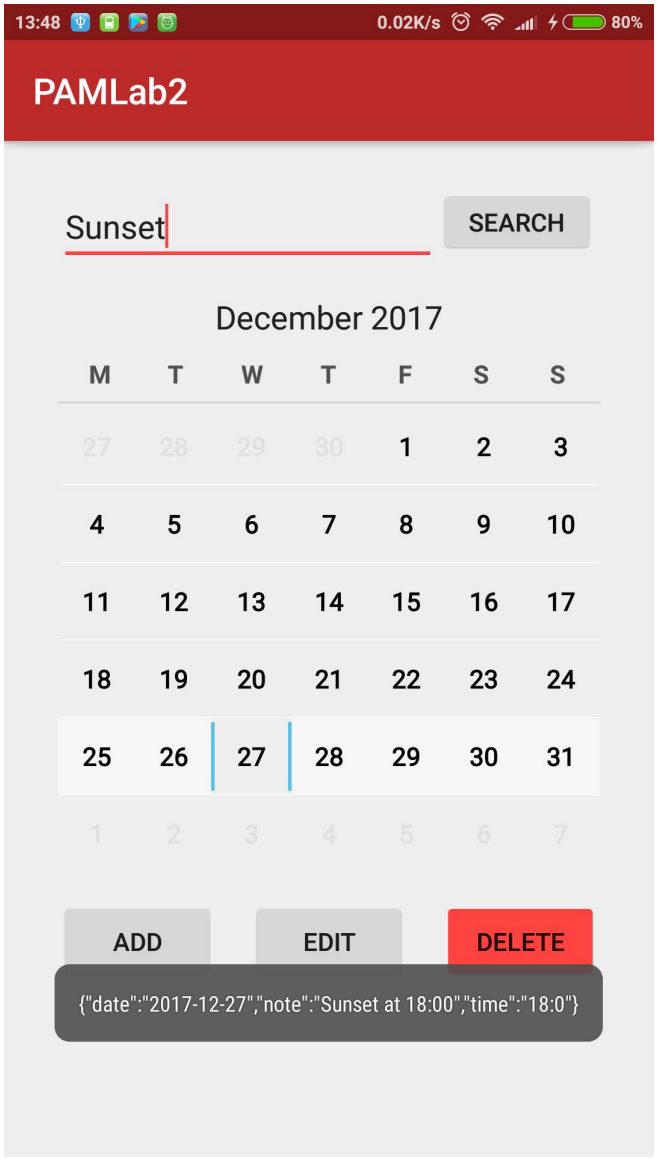

fig.5. Searching the keyword "Sunset" finds a previously created event, and sets the selected date to the detected entry

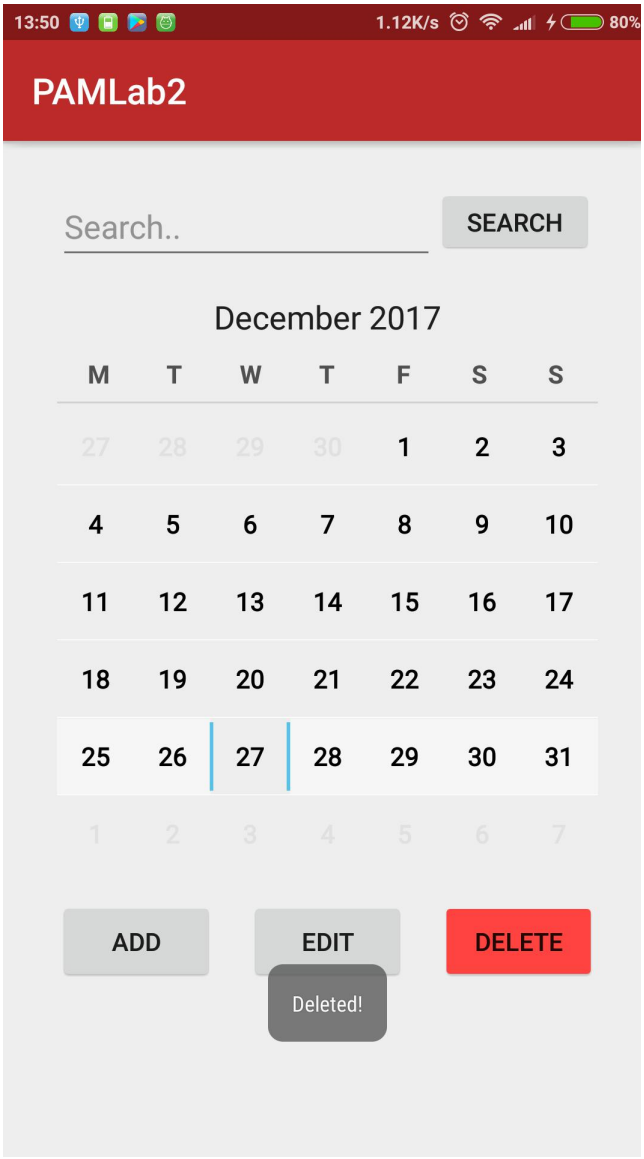

fig.6. Clicking delete, deletes the file/event. Trying to edit or search for it is now futile.

Conclusion

During this laboratory work, there was a lot of significant encounters with situations requiring learning and problem solving of levels exceeding personal ones.

The whole project took a few months to complete - which is, personally, a seriously stunning fact, because the progress was logarithmic. This shows, that the requirements for this laboratory work are overly extrapolated considering the knowledge the students have at this very point.

Regardless that, there was a good amount of learning involved and things like Buttons, text views, and other UI elements, that are the building bricks to an application design are now clear and obvious in use and development.

There's some other important concepts that were met in the current laboratory work.
Parsing data and working with alarms, and calendars. It is now an easy job to develop apps with this kind of looks and functionality.

The given laboratory work marks a new step towards the mastering of developing on a mobile platform. Plus, the designing process was a personal favorite. Choosing colors and making design decisions independently is what makes the creation process so enjoyable.

Bibliography

- **[1] CalendarView**
  http://abhiandroid.com/ui/calendarview

- **Handwritings from Mobile Applications Programming course** of Lector:
  prof.univ. I.Antohi.
  Chișinău: UTM
  2017

Annex

Code source:

https://github.com/StasBizdiga/PAM