Technical University of Moldova

Faculty of Computers, Informatics and Microelectronics

Department of Software Engineering and Automatics

Mobile Application Programming

Laboratory work #2

Organiser Mobile Application

Authors:
Alexandr Vdovicenco

Supervisor:
Sergiu Ciudin

Laboratory work #2

1 Topic and Objectives of the laboratory

”Organiser Mobile Application”

To create a Organiser app on the chosen platform. Applying strictly to the training and design of applications, what are required to be respected. At personal discretion, the application design remains and the API / Framework in which Laboratory 2 will be developed. Components and the structure of each Activity will be described below

UI Components The application will contain at least 3 basic activities that will be numbered in the paper as:

- MainActivity
  - Calendar View (custom or default)
  - Buttons (Add/Remove/Update)
  - Search (search by keywords)
- AddtActivity
  - Data/Time controller
  - Info TextBox
  - Buttons
  - and others (discretionary to the specific application)
- UpdateActivity - basically it’s one and the same activity from Add, just completed.

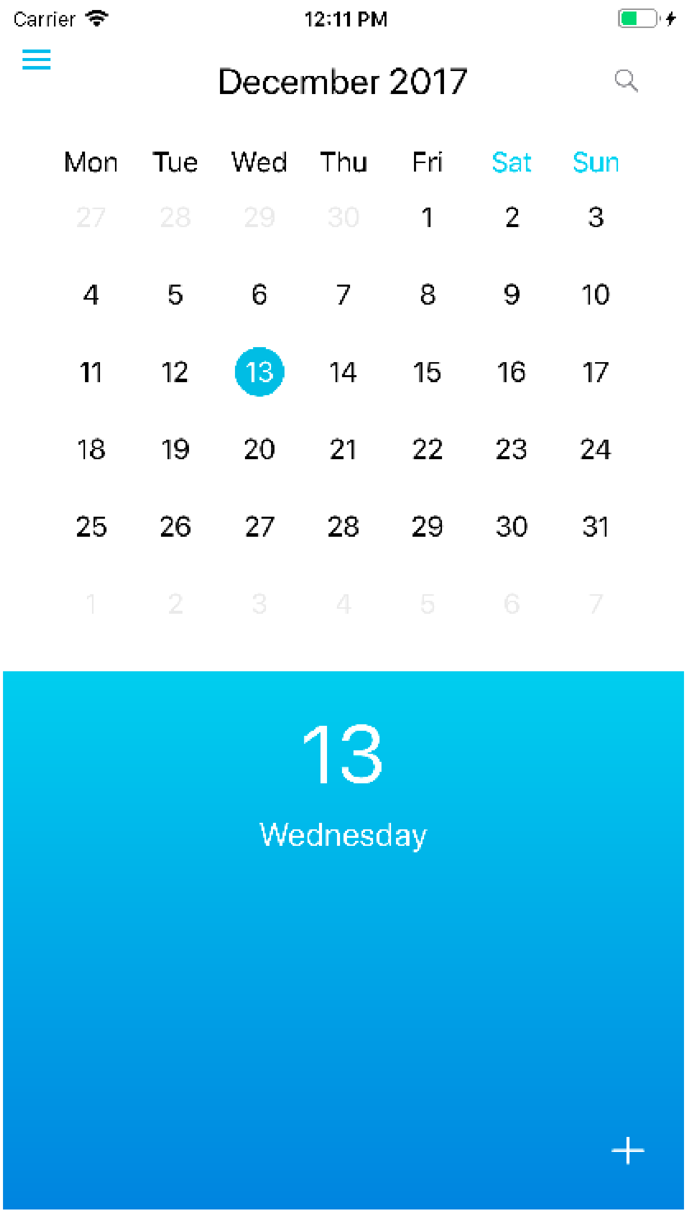
The operational data inside the application will be stored in XML file (s) that is left to discretion Personal. (keywords, XML Serialization).

Logical / Operational Component All events and actions of notification / signaling (audio / visual) undertaken in the Organizer will be treated in a separate service, which logically functional will extract the data from that XML file.

2 Laboratory work Task

The main task of this laboratory work was to create organizer application, with calendar view. The "iOS" doesn't provide a default system calendar view for using in personal application, and this is the reason why I had to use a library in order to provide a calendar view in application.

There are not so many libraries which provide a calendar view for "iOS", and I have chosen the "JTAppleCalendar" [1], which provide a fully customizing calendar view, and also is well documented.



Mon	Tue	Wed	Thu	Fri	Sat	Sun
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Figure 2.1 – Main View of Application

Because the library of which I used in order to get calendar view, provides full customization, I had found a template image for a calendar app, and perform this design in application.

Actually, implementation of design, and "UI" build have taken the biggest part of time at this laboratory work, and one of the reason is the fact that I build the "UI" of main view to be responsive for all device screens, and this was the hardest part of "UI" build, and took the biggest part of "UI" work.

When user adds new event to his calendar, he/she are redirected to new "View" of adding events, where is provided the time and date picker, and text inputs for adding name, and information about event.

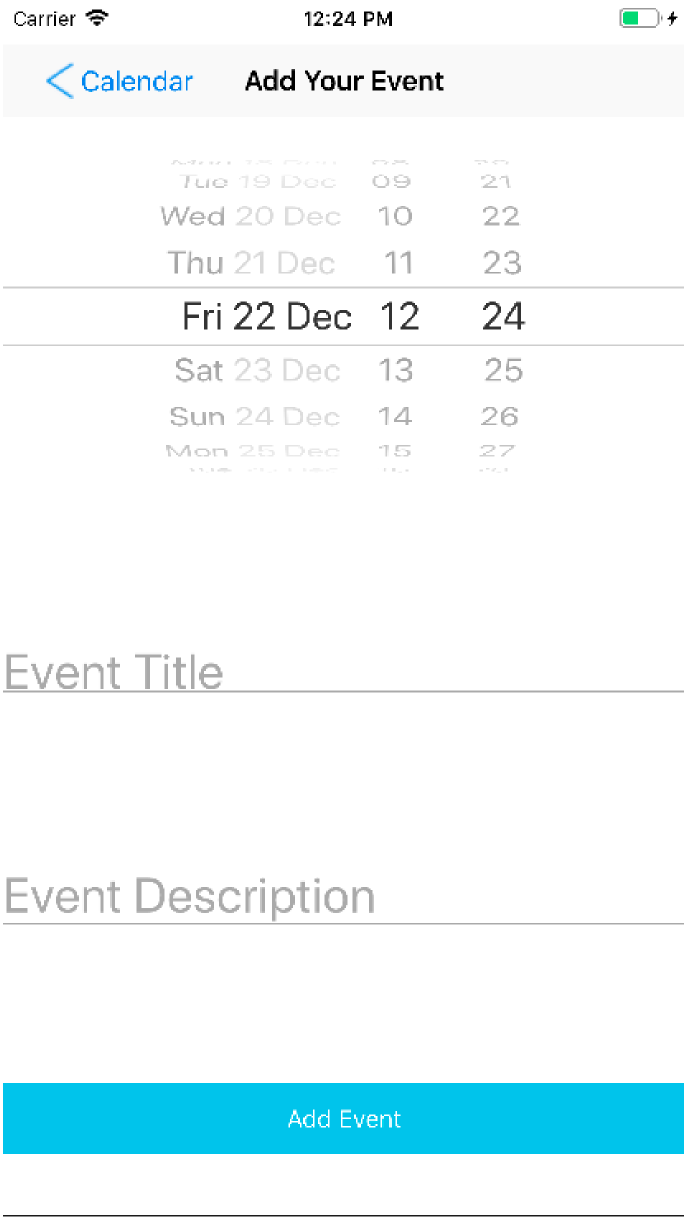
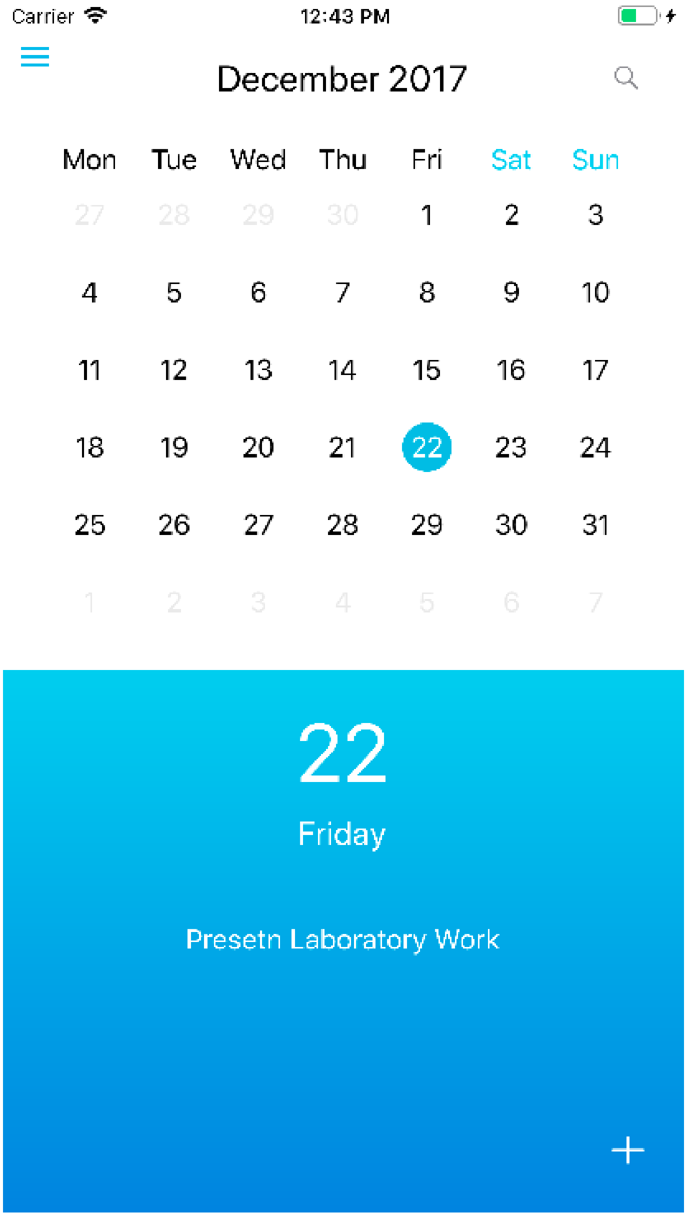


Figure 2.2 – Add event View

After pressing the button of adding new event, the new "JSON" object that contains information about event, such as date and time, event title and event description, is added to "JSON" array of objects which represents all events for one day. This "JSON" array is stored in "JSON" object with key being the date at which event should happen.

After the new "JSON" object is added to variable which represents application data about events, this variable is serialized into "JSON" file, from where it will be deserialized at next application start. In this way is performed save of user data in application.

After the new event is added, it's title is shown in the bottom part of application. If more events are added to one day, they are shown in the same order they was added, and there is the place for future improvement, events of the day to be sorted according to time when this event should happen, this means events, will be sorted chronologically.



December 2017

Mon	Tue	Wed	Thu	Fri	Sat	Sun
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

22

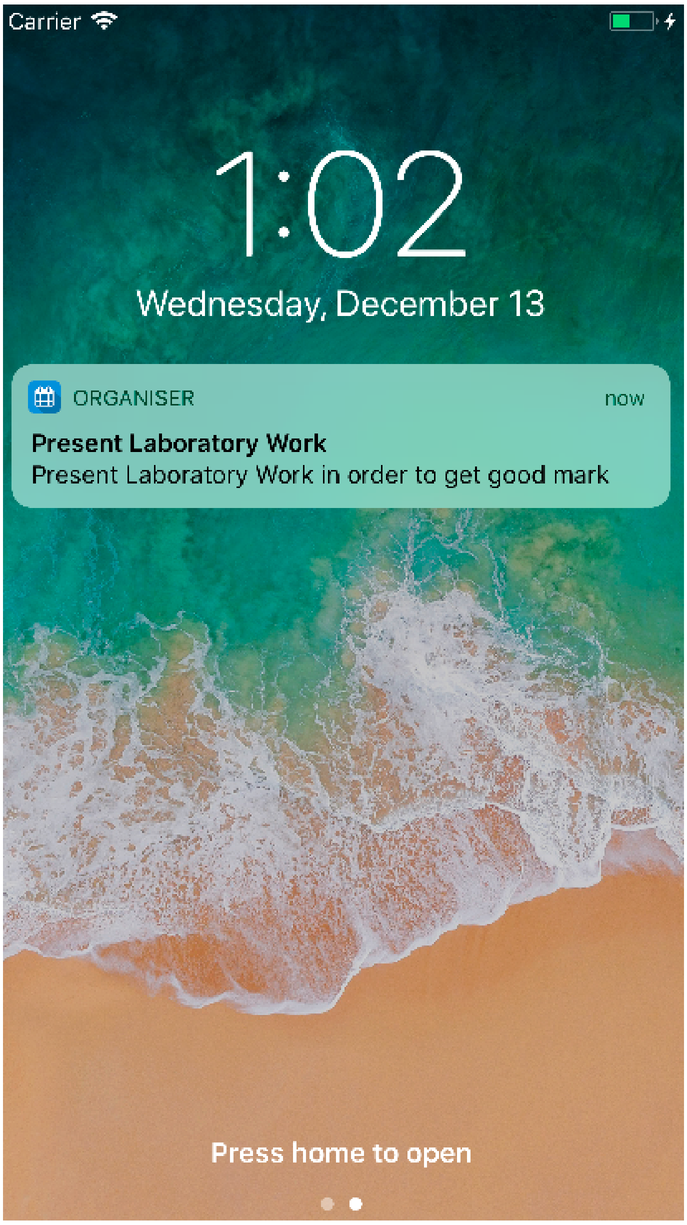
Friday

Presetn Laboratory Work

Figure 2.3 – Main view with event added

Event title also represent button which acts on long press, and when this event is triggered, user is redirected to the view of editing the event on which user pressed. View of editing the pressed event, is actually the same, but it is fill with data about event, this means that time and date picker are set to the event date and time, and also title and description field are also filled.

When an event should happen, application send a notification with custom sound, with longer play time, in order to notify user, that is time for a specific event. The reason why application doesn't provide an alarm clock, but only send a notification, is due the fact that "iOS" doesn't provide possibility to set an system alarm clock from an application.



ORGANISER

Present Laboratory Work
Present Laboratory Work in order to get good mark

Figure 2.4 – Notification receiving

Conclusions

As we can see, because ”iOS” doesn’t provide a default calendar view, to make an application which provide such functionality, require a hard work, for set-up library, and calendar view, which requires a lot of time comparing to ”Android” where calendar view is provided by default.

Also ”iOS” doesn’t provide an alarm clock like in ”Android”, and this is the reason why, an ”iOS” application has reduced functionality in terms of notifying user about event, because in ”iOS” is possible to provide only notification which has reduce functionality comparing to alarm clock.

All this issues make development of such application on ”iOS” platform, very specific and in some cases very hard, because you should deal with more custom elements which should be set-up, comparing to ”Android” platform.

References

1 https://github.com/patchthecode/JTAppleCalendar