Mobile Application Design 2019 Project

FIT Timetable

Team: **Alena Tesařová**, xtesar36
Petr Kohout, xkohou14
Dominik Harmim, xharmi00

FIT Timetable on Google Play: https://play.google.com/store/apps/details?id=com.tam.fittimetable

Project Goal

The main goal of this project is to create an application that can easily show students their *private timetable* so they do not have to sign in to the information system every time they want to know their schedule for the next day. The next problem of the private timetable on the web is a chaotic arrangement of courses – we can see the same course several times (for different rooms and different groups), we never know the regularity of the course (because of the week view), and courses are mixed with occasional events such as exams or labs.

Used Technologies

- We were using *Android Studio* as an IDE. It is quite easy to use, but on the other hand, it has really extensive and useful features and during this project, we gained many experiences with this IDE. For testing the application in multiple devices, we were using builtin *Android Virtual Device (AVD)*.
- The main programming languages of this project are *Java* and *Kotlin*. They are compatible and they can be used simultaneously. We were using both of them.
- When we were publishing the application, we were using *Google Play Console*.
- During the development process, we were using *Git* version control and *GitHub* as a remote repository.
- We used the online tool *Figma* for creating a logo.

Used Resources

Documentation:

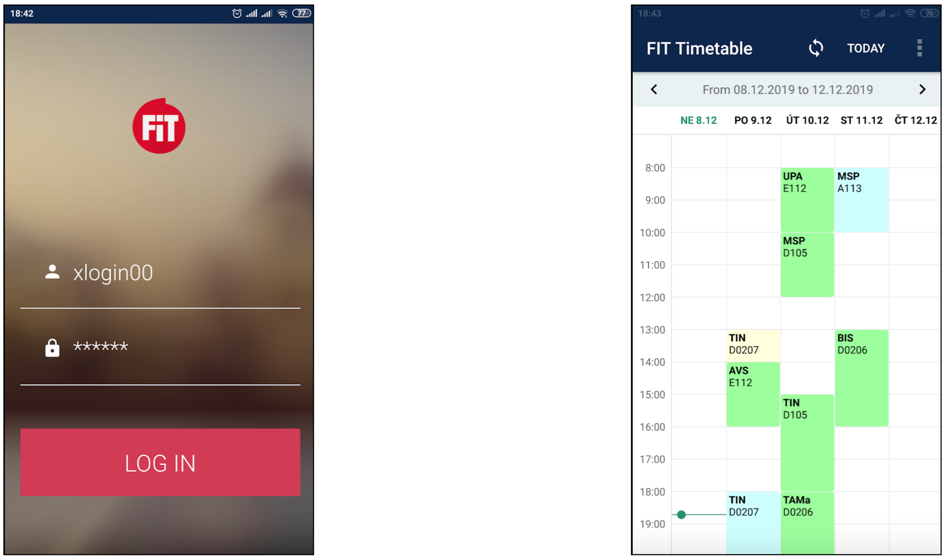
- Android documentation – https://developer.android.com/docs.
- Java documentation – https://docs.oracle.com/javase/7/docs/api/overview-summary.html.
- Google Play Support – https://support.google.com/googleplay/android-developer.

Libraries:

- Android Week View for displaying the timetable – https://github.com/thellmund/Android-Week-View.

Most Important Achieved Results

The most important is *singing into the application* and *displaying the timetable*. These two activities can be seen in Figure 1. The signing activity is important because a user is authenticated to the *information system of FIT VUT*. Anyway, the main part of the application is the *timetable view*. Courses are downloaded from the information system. They are parsed and appropriate information is extracted and displayed. Courses in this view may be removed and there is also a possibility to actualise it from the information system again.



xlogin00

******

From 08.12.2019 to 12.12.2019

NE 8.12 | PO 9.12 | ÚT 10.12 | ST 11.12 | ČT 12.12

UPA E112
MSP A113
MSP D105
TIN D0207
BIS D0206
AVS E112
TIN D105
TIN D0207
TAMa D0206

(a) The login screen

(b) The main timetable screen

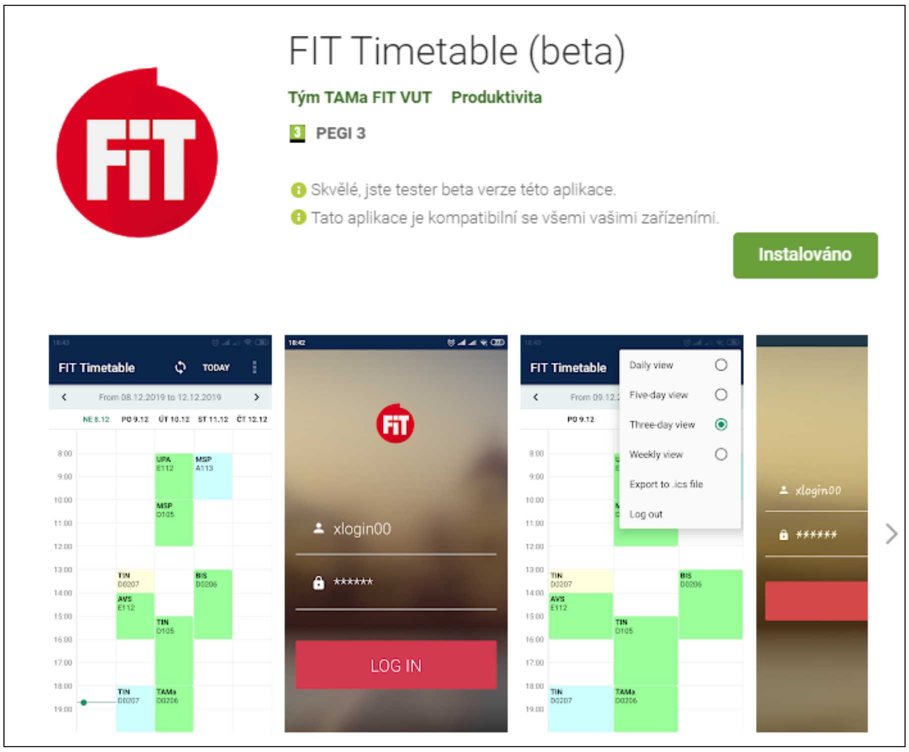
Figure 1: The main screens of the application

Furthermore, important achieved result is that we published the final application on *Google Play*, see Figure 2. Firstly, we published it as a beta version. And later, when we fixed some bugs, we published it as a stable version.

Controlling of Created Program

Our application is very simple to use. To sign up it is necessary to fill in a login and a password and press the *Log in* button (Figure 1a). After signing up, the application shows the timetable for the actual week and weeks can be easily change by swiping the screen or by tapping on arrows next to the date (Figure 1b). Next, there is a button for actualising the timetable from the information system (it downloads a current view of the private timetable) and there is a menu with several options such as:

- daily view,



FIT Timetable (beta)

Tým TAMa FIT VUT Produktivita

PEGI 3

Skvělé, jste tester beta verze této aplikace.

Tato aplikace je kompatibilní se všemi vašimi zařízeními.

Figure 2: The published application on Google Play

- three-day view,
- five-day view (default),
- weekly view, and
- export to *.ics* file (for export to different applications).

Experience with the Selected Platform

We were programming in *Android Studio* using GitHub to share the code. As new users of Android Studio, we were pleased to work with this tool – it helps with syntax, orientation in a code, downloading needed libraries, etc. To simulate the code, we were using an emulator with different devices. We struggle the most with logging to the information system using VUT and FIT certificates (using `TrustManagerFactory`, `CertificateFactory`, SSL context) — we could not find out how to write the certificate to the mobile storage and how to read the context on a device.

Distribution of the Work in the Team

- **Alena** designed and created a user interface. She dealt with a graphics framework. She also led the team. Finally, she linked frontend with the backend API.

- **Petr** created a backend API for downloading and parsing the timetable from the FIT VUT information system into appropriate data structures. He also created an API for authenticating to that system.
- **Dominik** created a feature that allows exporting timetable to different applications. He also tested the application and fixed some frontend bugs. At last, he deployed the application to the Google Play.

What Was the Biggest Challenge

As it was already mentioned above, one of the biggest challenges was to manage *authenticating* to the FIT VUT information system because we had to work with suitable *certificates*. Quite time-consuming was *parsing HTML pages* from the information system because there is no documentation how these pages could look like. Further, it was challenging to deal with the user interface. In particular, it was not easy to create *timetable view*. In the end, we used a framework for that.

Experience Gained From the Project

We learned how to design a user interface for mobile devices, particularly for the latest Android devices. Furthermore, we learned how to use Android Studio for Android development, we improved our Java skills, and we learned the basics of Kotlin language. We also gained some experiences of parsing HTML pages and using SSL certificates. Finally, we learned how Android applications deployment works.

Autoevaluation

Technical Design (95 %): At first, we created *backend API* for authentication to the information system and downloading and parsing timetable from that system. Then, we designed and created a user interface and we linked it with the backend API. For the frontend timetable view, we selected appropriate framework.

Programming (80 %): We used the *object-oriented paradigm*. The code is split into several classes. Some classes are written in Java and some in Kotlin. Methods in classes are sensible named so the overall code is quite readable. In run-time, exceptions are caught and appropriate messages are shown to a user. The backend is general and it can be used in other applications as well.

Usability of the Created Solution (90 %): The application can be easily used by students of the FIT VUT. Some of the students already installed and tried the application and they were satisfied with it.

Use of Resources (70 %): We used the *Android Week View* framework for displaying the timetable. During the development process, we were using Android and Java documentation. For downloading and parsing the timetable, we used handy Java libraries.

Time Management (75 %): In the beginning, we planned to implement more features and extensions but finally, we did not have time for it. However, we implemented core functionality on time and we had enough time for testing and fixing major bugs.

Team Cooperation (90 %): We were seamlessly communicating via *Messenger*. We split the work between team members at the beginning of the semester and we also had several personal meetings, primarily before project workshops.

Chances of Publishing the Application (100 %): The application is already published on Google Play. It is available on the following link `https://play.google.com/store/apps/details?id=com.tam.fittimetable`.

Overall Impression (80 %): The main goal was achieved. The application is useful for students of the FIT VUT and it is published. Source codes are available on GitHub so it can be extended in the future. We gained many useful skills and experience, as it is described above.

Five Main Questions of TAM

What drives this sector of IT?

This sector of IT is driven by the *increasing number of mobile phone users*. More and more people own a mobile phone and nowadays almost everyone has one. People carry their phones always with them so they use them for communicating, keeping notes, for entertainment, for paying, for controlling their smart homes, etc.

What it will be like in five years?

It will probably be more-less the same as nowadays. Of course, there will be significantly more applications. Maybe, there will be new applications related to *virtual reality* and the *internet of things*.

What slows it down, speeds it up?

The factor that speeds it up is that there are more and more powerful devices with *efficient hardware*. In the other hand, currently, there are many different versions of operating systems. Moreover, there are two different widely used operating systems (iOS and Android). So this fact slows this sector down.

What ideas are dead (though they appeared great once)?

The *Windows Phone operating system* is dead, although it had potential and it was quite successful from the beginning. Another promising idea which is now dead is a *hardware keyboard* on smartphones.

Where do new ideas come from?

The most new ideas come from *people's everyday demands and needs*. New applications are based on ideas of how to make *people's life easier*.

Recommendation for Assigning Future Projects

The *project workshops* are good. The acquired *feedback* is often very useful. It was also very nice that we had a chance to pick our *own assignment* and even *own platform*, so we could learn exactly what we wanted to.

One disadvantage of this project that we have to mention is *late information* about when and how to send slides to project workshops. These dates and instructions maybe can be available beforehand for the whole semester.

Recommendation for Future Students

In such a project, think of a *really minimalist* and *useful* application and do it properly. Probably, you will be in time pressure and there will not be enough time to finish your project. At the beginning of this course, you might think that your selected assignment is simple and that it will be easy to design and implement it. But try to truly *think of the problem*, its *design*, and *development*. Finally, pick something *simple* and focus on it from the beginning. Focus on design, implementation, testing, and deployment.