Application of a new technology stack in the development of the mobile app «Schedule of classes»

A A Popov, E M Gritsenko, M M Satsuk, A A Artyshko and D E Mironov
Reshetnev Siberian State University of Science and Technology, Krasnoyarsk, prospect "Krasnoyarsky rabochy" 31, Russia

E-mail: tolnbms@yandex.ru

Abstract. The task of improving the effectiveness of activities aimed at prompt receipt of information by students of Reshetnev Siberian State University of Science and Technology is solved by using the developed mobile application «Schedule of classes». Due to the heavy load on the University's unified server, there are temporary delays in accessing the main site, which makes it inconvenient to use it. In this regard, there is a need to solve this problem by bringing the described process to partial autonomy. This approach will allow you to combine high performance due to the speed of the mobile operating system and expanding the functionality of both the application itself and the means of sending information from the management of the Institute to students. The purpose of this work is to study and analyze existing mobile applications of other universities, their advantages and disadvantages. Based on the research, develop a mobile application, agreeing with the Directorate of the Institute of Informatics and telecommunications tasks. This article presents the main aspects of the design and software solution. The software solution is implemented in the Android Studio integrated development environment in the Dart programming language, using the Flutter SDK.

1. Introduction
Currently, more and more people are actively using mobile devices around the world – smartphones, tablets, and other gadgets with the ability to access the network. In this regard, it becomes obvious that the use of specialized applications for specific tasks is one of the priority tasks of software development, while the integration of various tasks into one single information portal on the Internet is inefficient for the following reasons:

- Too much information load makes it difficult to get it.
- Since all the services used are contained on the same portal, as a result, due to a large flow of users, the system cannot cope with loads.
- There is a high risk of DDoS attacks, which can lead to complete paralysis of all users and employees of the portal.
- Adapting the entire information portal to mobile devices is an extremely resource intensive operation.
- Lack of ability to transmit and receive information promptly.
Universities operate in the context of a large volume of constantly changing information that needs to be quickly analyzed and appropriate decisions made. In this regard, it became necessary to design and develop a mobile application that allows you to lead to partial autonomy of some of the services used in the unified information portal of SibSU University [1].

Research and analysis of existing solutions show that existing solutions have a number of disadvantages. The following existing applications were used as an example: «The schedule FINEC», «the TIU Schedule», «Schedule BSU». Each of these apps is designed to serve the original schedule to students and teachers. All of them are up-to-date and working, but there are a number of disadvantages inherent in most such applications.

The following common problems stand out among all of them:

- Group selection is provided from the drop-down list, without the possibility of manual search (which makes it difficult to find the desired group).
- There is no offline access to the app.
- Inconvenient / not intuitive UI (user interface), which negatively affects the daily use of the app.
- There is no functionality for instant display of today's schedule at the time of application launch.
- There is no functionality for notifying users about schedule updates.
- It is not possible to view current news information about the University.
- Complete absence of the University's campus map.
- Part of the app's functionality is in English, without the ability to change the localization.
- Display of dual classes without a mark on the division into subgroups.
- Most of these applications have only one basic operating system, and often there are no alternative operating systems (for example, Apple iOS).
- Some apps do not allow you to view a specific teacher's schedule.

In addition to domestic developments, foreign technologies and principles of scheduling in universities were analyzed. The system of higher education in the United States is very flexible: students choose their own subjects (although there are mandatory ones), make up a schedule, can change their specialty, and so on. At the same time, one of the most important features of training is practice. All the material learned during the lectures is then studied in practice, which makes it possible to get working experience before the end of training.

Based on the analysis, it can be concluded that the educational process in higher education institutions is formed on the basis of block training with a two-week cycle of alternating weeks. In this regard, there are some strict restrictions: classes in disciplines must be held in the same audience, at the same time, throughout the entire period of study.

Based on the logic described above, mathematical models of the problem of scheduling University classes offered by foreign and Russian researchers were also considered [2].

The process of design, documentation and development of the application considered in this article provides for the main disadvantages of existing analogues, and also takes into account the technical features of the possibility of merging two systems (the University's information portal with the application itself) [3].

2. Methods

All modern information systems are working with information organized in the necessary structure - a database. Databases provide storage and provision of information and are a set of data organized according to certain rules that include General principles for describing, storing, and working with data.

The mobile application under development includes a large amount of information: text, graphics, links, tables, and other types.

A key stage in the implementation was the need to take into account the fact of a large load (on the part of users), as well as the fact of a large amount of information in the database. In order to optimize
the process of loading and processing data, The MySQL 5.7 DBMS was selected, which allowed to ensure the system's performance [4]. Currently, the design diagram of the described mobile application database is presented, and the structure, relationships, and objects are defined (figure 1).

Figure 1. Diagram of the database implemented in this application.

The functionality for the mobile app user is presented in the form of a design solution using use cases (figure 2). This aspect allows you to understand what application functions and information are available to the end user and what actions can be performed [5].

Logical aspect—allows you to define the functional requirements of processes. It specifies the logical relationship between classes of process elements. To build models, use class and state diagrams.

Component elements—pay attention to the composition of process elements and their distribution when creating an information system (IS). Models in this aspect are built using a component diagram. It contains information about process elements and software.

Commissioning—this aspect allows you to show the process diagram in relation to the hardware in the is. Only one diagram is used for building models: the topology diagram [6].

In addition, at the design stage it is necessary to provide a system of access rights. This system allows you to describe sets of rights that correspond to the positions of users or type of activity. The structure of rights is determined by a specific application. To implement the restriction of access rights in application solutions, special objects are designed - Roles. The role determines what actions, on which metadata objects, the user acting in this role can perform. This aspect makes it possible to understand how the participants in the process act and thereby determine their interaction and impact on the process. To build process models within this aspect, Use-case diagrams, sequence diagrams, collaboration diagrams, and action diagrams are used.
Figure 2. Diagram of usage cases for the app user.

The operation of any information system and any applications cannot function without administration. The administrator is responsible not only for the trouble-free operation of the system, but also for maintaining updates and up-to-date information. In this regard, the administrator functionality and system usage options for the administrator were developed (figure 3).

The analysis of requirements to the schedule of training sessions, the decision was made necessary in the development of the algorithm, which would allow for the expansion of the list of requirements schedule of classes, as well as the possibility to adjust execution priorities of the individual requirements when scheduling.

Creating a class schedule begins with entering data about the training load. The algorithm also requires data about audiences, if they were not entered earlier, they are entered. Depending on the options for implementing the user interface, lists of teachers and groups can be generated automatically, based on data about the training load, and then supplemented with missing data, or they can be entered manually.

The sequence of applying the basic algorithms is determined by the complexity of the work. depending on this complexity, attempts are made to include this work in the entire schedule on different days and different pairs.

For algorithmic support of this balance, the following sequence of application of basic algorithms is laid down. For disciplines with a high level of complexity, the selected local algorithm is applied in the sequence: Wednesday, Thursday, Tuesday, Friday, Monday, and Saturday. For disciplines with a low complexity index, the sequence of applying local algorithms is different: Saturday, Monday, Friday,
Tuesday, Thursday, and Wednesday. For disciplines with an average complexity index, the sequence of applying local algorithms is as follows: Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday.

![Use case diagram](image)

**Figure 3.** Use case diagram for the administrator.

3. **Software implementation**

For software implementation, the Android Studio IDE was used as the main development environment. This is a development environment for the popular Android operating system. In each new version of Android Studio, the developer adds increased functionality, optimizes processes, and more. Included with the IDE is an emulator that checks the correct operation of already written utilities and applications on different configurations.

Programming was done in the Dart language, a programming language created by Google. Dart is positioned as a replacement / alternative to JavaScript. One of the language's developers, Mark Miller, wrote that JavaScript "has fundamental flaws" that can't be fixed. This is why Dart was created [7]. In addition to the main libraries of this language, the SDK framework for cross-platform development of Flutter was used. This is an open source SDK for creating mobile apps from Google. It is used for developing apps for Android and Apple iOS, and it is still the only way to develop apps for Google Fuchsia.

The server-side technology for the application was PHP 7, a General-purpose Scripting language that is intensively used for developing web applications. Currently, it is supported by the vast majority of hosting providers and is one of the leading languages used for creating dynamic websites [8].

In order to start working in the system, immediately after launching the app, you must select your group, or enter the full name of the teacher. This is done by entering the first letters of the group name / full name, after which you need to select the group or teacher, respectively, from the list in the drop-down list (figure 4).
After the user has selected a group, the schedule for the corresponding group opens (figure 5). This information is stored in the local NoSQL database, which allows you to view the schedule of your group even offline, without a network connection. If you need to update (download the current schedule) - just swipe your finger from top to bottom, and then download from the official website of the schedule.

As you can see in the image, it is possible to move through different weeks, viewing the entire schedule of any of the groups and any teacher. For ease of use, the current day of the week is marked in the upper-right corner, and opens automatically when the app is launched.

As you can see from the example, there may be repetitions in the list of activities. This is due to the fact that each lesson is a separate object that must be placed in the schedule. Duplication occurs due to the fact that two classes of the same type are held during the same week.

In addition, there is a “Exams” tab that displays information about exams for a specific group. In future updates, we plan to add a timer that would remind your user of the upcoming exam or test.

There is also a separate side menu that contains a list of the main functions and features of the app (figure 6). this list includes: events (University), building map (campus map), and settings of the app itself.

The events section includes all University news published on the official website. If there is a check mark for notifications in the settings, each publication will receive a push notification.

The hull map is a use of the Google Maps service imported into this app. The interface is a map of the city of Krasnoyarsk with the University buildings marked on it, as well as your personal location. This item is especially relevant for first-year students.

The settings section currently only has the function to enable/disable notifications about new events. In the future, this section will be expanded and adapted to specific tasks.
In addition, there is a separate function that allows you to select any of the calendar days to view the schedule for a specific day. This will allow each user to plan their to-do list in advance. The only drawback of this feature is that the schedule is only available for this semester.

As a result of processing data about the training load of threads, a list of classes is generated.

4. Discussion
The advantages of the developed mobile app include the following:

- Fully auto-updated schedules from the University’s website (https://timetable.pallada.sibsau.ru/timetable/).
- Application performance in comparison with applications created in native languages. [9].
- Usability (the ability of a product to be understood, studied, used and attractive to the user under specified conditions (ISO/IEC 25010)) [10].
- Control the work of the various versions of the application in real time using Firebase Google [11].
- Offline access to schedules, news and upcoming events.
- View the current schedule, even when the official schedule site is not working.
- View all buildings of the University from your personal location on the map using GPS (offline access is available).
- Beautiful and responsive Google Material app design for different device screens.
- Support for earlier versions of operating systems Android 4.1, Apple iOS 8.
- Informing users about schedule updates or new news in the app.
- There is support for enabling/disabling channels of information.
5. Conclusion
As a result, a fully functional mobile application was developed and implemented, the popularity and demand for which reached just under one thousand users in a short period of time, among which there are both students and teachers of the University.

The development does not stop there, in the future it is planned to implement a separate module in the same application for applicants, with the possibility of having a personal account that would display all information about the admissions Committee, including its current position in the ratings [12]. In addition, a platform is being prepared for porting this software for mobile devices on the Apple iOS operating system, which will increase the demand and popularity of the described product.

Further support and development of this project will help to solve many problems and shortcomings that exist in the current information portals of Reshetnev Siberian State University of Science and Technology.

References
[1] Beregovyh Yu V, Vasiliev B A and Volodin N A 2009 Class Schedule Algorithm Artificial Intelligence 2
[2] Bezginov A N and Tregubov S Yu 2011 Complex of algorithms for building a University schedule 2: Structure of data representation and algorithm of the reference solution Bulletin of the Baltic Federal University named after I. Kant 10 93-102
[3] Donetskov A M 2008 Automation scheduling training sessions in the University all-Russian scientific-technical conference “High technologies in instrument and mechanical industries and development of innovation activities at the University” 2 (M: publishing house of the MSTU. N. E. Bauman) 98
[4] Carvin B 2012 SQL: database Programming Typical errors and their elimination 336
[5] Abramova O F, Kaplya V I and Lyasin D N 2013 Guidelines for laboratory work on the topic “Basic information about UML and BOUM:Diagrams of use cases” (Volgograd: Volgograd state technical University) p 49
[6] Leonenkov A V 2006 UML: Object-oriented analysis and design using IBM Rational Rose 568
[7] Traub J, Vasilkovsky G and Vozhnyakovsky H 1988 Information: uncertainty, complexity Jonathan McCallister Mobile App Made Simple: The Ultimate Guide to Quickly Creating 2014 68
[8] Koterov D V and Simdyanov I V 2016 PHP 7 in the original 1073
[9] Rafgarden Tim 2020 The Perfect Algorithm. Greedy algorithms and dynamic programming (St. Petersburg)
[10] Alan C Robert R 2009 Alan Cooper on the interface - fundamentals of interaction design (SaintPetersburg: Plus Symbol) p 688