

# Development of CRM system with a mobile application for a school

Nataliia I. Cheboksarova<sup>1</sup>, Tetiana A. Vakaliuk<sup>1,2,3</sup> and Iurii M. Iefremov<sup>1</sup>

<sup>1</sup>Zhytomyr Polytechnic State University, 103 Chudnivsyka Str., Zhytomyr, 10005, Ukraine

<sup>2</sup>Institute of Information Technologies and Learning Tools of the NAES of Ukraine, 9 M. Berlynskoho Str., Kyiv, 04060, Ukraine

<sup>3</sup>Kryvyi Rih State Pedagogical University, 54 Gagarin Ave., Kryvyi Rih, 50086, Ukraine

## Abstract

The article substantiates the need to develop a CRM system with a mobile application for a school. The existence of their educational environment, which will provide interaction between pupil and teacher in a school, is a topical issue today. Relationship management can be implemented in the form of a CRM system. For a deeper understanding of the research problem, certain analogs are considered, their advantages and disadvantages are given. The work aims to design and develop a CRM system with a mobile, cross-platform, application for a school. As a result, the architecture of the software package was built, the choice of tools was substantiated, the use and structure of the system were determined, the object-oriented structure of the system was designed, the data storage structure was developed, the system operation algorithms were designed and implemented, the system installation and administration procedure the procedure for working with the software package.

## Keywords

CRM system, mobile application, school, learning environment, development

## 1. Introduction

One of the main issues in the management of educational institutions, including schools, is to increase the level of educational and methodological work of a particular institution. A digital educational platform would allow the effective use of available educational institution resources, and students and teachers would be allowed to use modern technologies in practice.

The main advantages of such technologies are: performing various types of educational work, control, and evaluation of knowledge online; open learning environment for subjects of the educational process, etc. All this leads to the fact that the use of various digital technologies provides an opportunity to: keep electronic journals; use online services for the learning process; conduct correspondence, testing, and assessment of knowledge online; communication, etc.

---

CS&SE@SW 2021: 4th Workshop for Young Scientists in Computer Science & Software Engineering, December 18, 2021, Kryvyi Rih, Ukraine

✉ nataly.cheb70@gmail.com (N. I. Cheboksarova); tetianavakaliuk@gmail.com (T. A. Vakaliuk); eyuri72@gmail.com (I. M. Iefremov)

🌐 <https://sites.google.com/view/neota> (T. A. Vakaliuk)

🆔 0000-0001-6517-6048 (N. I. Cheboksarova); 0000-0001-6825-4697 (T. A. Vakaliuk); 0000-0002-1249-5560

(I. M. Iefremov)



© 2022 Copyright for this paper by its authors.  
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

Given the pace of modern life, many people seek to save their time by using mobile applications. But at the beginning of their design, there is a choice: to choose a ready cross-platform framework or to develop separate applications for each end platform. Both solutions have their disadvantages and advantages. For example, cross-platform solutions significantly speed up the development process, because they meet the principle: deliver - as much as necessary and write once. And native applications have much faster performance than universal ones. However, they require separate construction of interfaces and writing application functionality for each of the platforms.

The existence of their educational environment, which will provide interaction between student and teacher in a school, is a topical issue today. Relationship management can be implemented in the form of a CRM system.

CRM (Customer Relationship Management) is literally means “customer relationship management and refers to all strategies, methods, tools, and technologies used by business to develop, retain and attract customers” [1].

## **2. Theoretical background**

In recent years, many studies by various scholars have focused on the design and use of the digital learning environment [2, 3, 4, 5, 6]. In particular, Uchitel et al. [7] considered the implementation of future agricultural engineers’ training technology in the informational and educational environment. Munk et al. [8] researched cloud-based educational platforms, Vlasenko et al. [9] raised the issue of management of online platform development and support process.

Sultanova et al. [10] investigated the potential of Google Classroom web service for lecturers of higher educational establishments under pandemic conditions, Shyshkina [11] proposed service models of the cloud-based learning environment of the educational institution, Popel and Shyshkina [12] researched the areas of educational studies of the cloud-based learning systems.

Soroko et al. [13] considered educational electronic platforms for STEAM-oriented learning environment at general education school, Osadcha et al. [14] conducted the review of the adaptive learning systems for the formation of individual educational trajectory.

Some scholars have attempted to analyze the possibilities of integrating business simulations software into the learning environment of technical university [15]. In addition, Vakaliuk et al. [16, 17] considered it appropriate to research the state of ICT implementation in institutions of general secondary education.

## **3. Related work**

For a deeper understanding of the research problem, consider certain analogs, in particular, the application “School – Learning Assistant”, the interface of which is presented in figure 1. In this figure, it is possible to view the list of classes for a specific day, the schedule of the week, and the list of added information about other students and teachers.



**Figure 1:** School application interface – Learning Assistant.

Analyzing the application revealed significant shortcomings, in particular: the lack of ability to specify the exact time of class and make the so-called “windows” (because it is not always necessary for all classes to take place one after another), lack of functionality designed only for teachers, and no the possibility of interaction between teacher and student, in addition, the student must enter all the information about the lesson, schedule, classmates themselves.

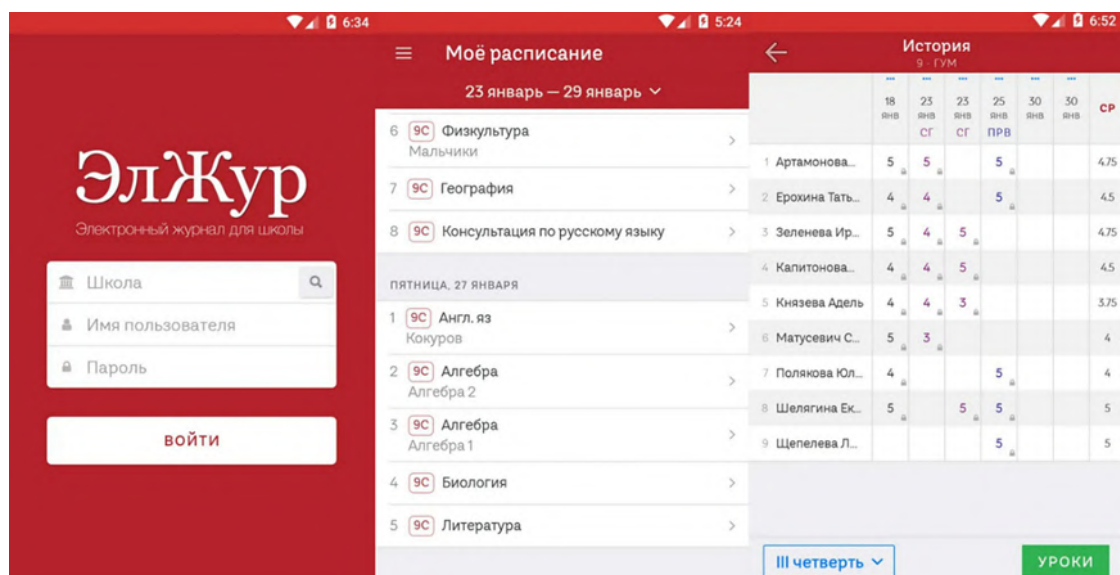
Among the advantages are the convenience of viewing the schedule for the week, the ability to save homework, save information about classmates. You can also note the intuitive interface, so understanding the work of the application was quite easy.

Next, we will consider the application ElZhur.Uchitel. The purpose of this appendix is a journal for the teacher, which shows a list of subjects and the journal of student assessments (figure 2).

A significant disadvantage is that the journal is designed only for the teacher and its data are not displayed to students. In general, there is no functionality for students, so, as in the previous application, there is no interaction between users.

Among the advantages can highlight the convenient division of the list of the magazine into academic quarters, as well as the calculation of the average score. It was also appropriate to use one color scheme and conciseness of the whole design.

Next, consider the application Weeklie. After analyzing the application, significant shortcomings were identified, including the lack of educational material on using the application with such functionality, the lack of functionality designed only for the teacher, and the lack of interaction between teacher and student, in addition, the student must enter all information about classes, schedules. Functionality is not enough for easy use.



**Figure 2:** The interface of the application ElZhur.Uchitel.

Among the advantages is the ability to specify the exact time of the lesson, the ability to save homework, user notification system (figure 3).

Moyklass.com is a CRM system for educational institutions, including schools, that allows you to perform the following functions: individual and group classes, view lesson schedules, keep a log of visits, homework, and grades, students' birthdays, students' accounts, downloads documents, financial accounting, reports on various forms of work, mobile version, etc (figure 4).

bitrix24.ua is the CRM system for educational institutions. This is a system for automating the work of various educational institutions. CRM is a convenient set of tools for all actors in the educational process: students, teachers, school leaders, etc. This system provides the following features: viewing class schedules, video communication, automatic creation of student rankings based on analysis of attendance and performance, maintaining digital documentation of the school, integration with special systems for online testing of students (figure 5).

After getting acquainted with the analogs and their analysis, we can conclude that each software application is significantly different from its competitors.

The biggest disadvantage of all the above applications is the lack of interaction between students and teachers. All data is stored locally by the user and does not affect the system as a whole. Benefits include easy to use schedules, track homework, alert system, and statistics collection.

The work aims to design and develop a CRM system with a mobile, cross-platform, application for a school.

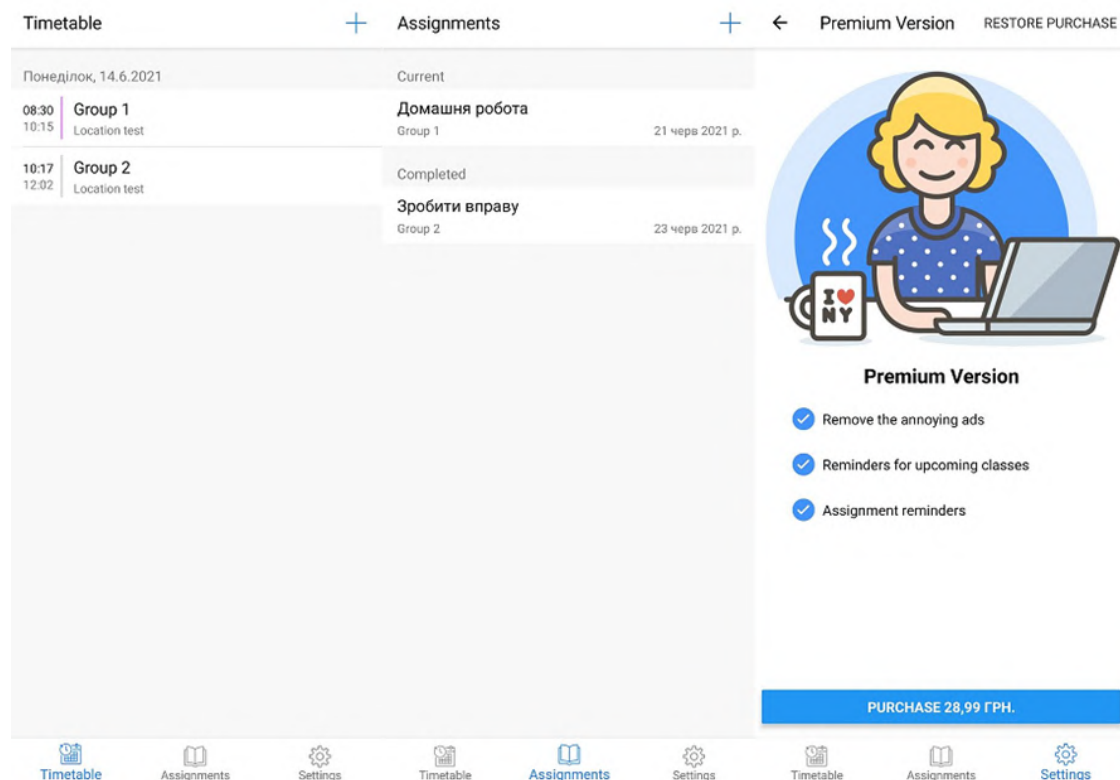


Figure 3: Weeklie application interface.

## 4. Results

### 4.1. Construction of software architecture

The application architecture can be considered the foundation that holds everything together and supports the codebase as it grows. If you have a good foundation, it becomes easier to make changes and add new things. The architecture uses design templates to solve problems effectively. Therefore, it is necessary to choose the design templates that best suit the problem to be solved.

Client-server architecture is the dominant technology in the creation of distributed network applications. It provides data exchange and interaction between developed software products. Client-server architecture can be defined as the concept of an information network in which the bulk of its resources are concentrated in servers serving their clients. The model of client-server interaction is determined primarily by the division of responsibilities between the client and the server.

Clients and servers of this architecture operate in parallel and are independent of each other. More than typical is the situation when one server simultaneously processes requests from different clients; on the other hand, the client can access one server or another. Clients need to know about available servers but may have no idea about the existence of other clients.

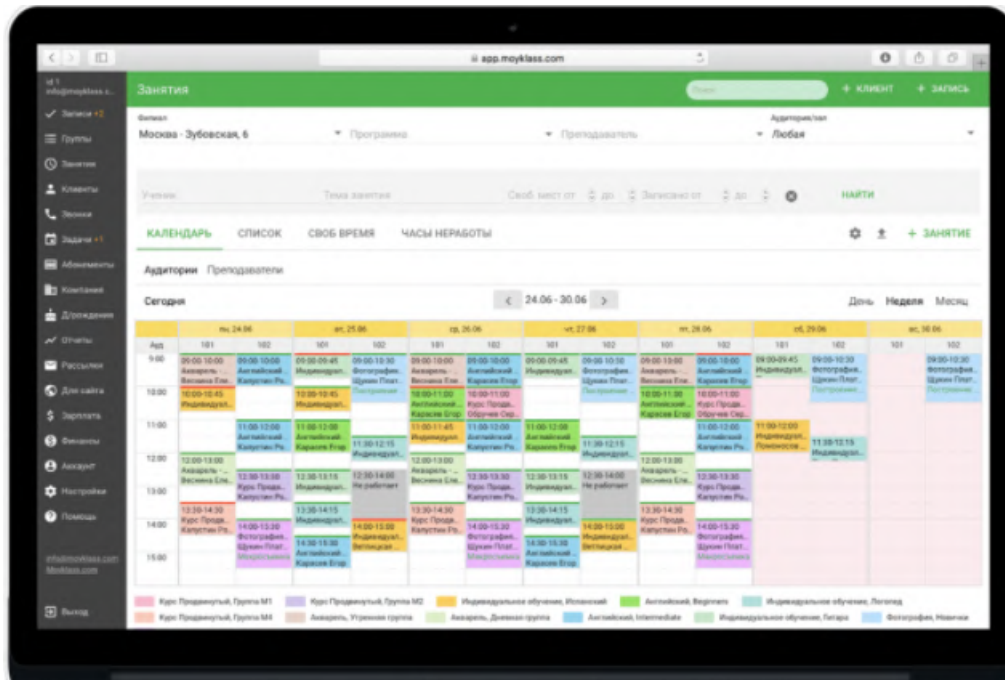


Figure 4: Moyklass application interface.

The advantages of this architecture include the reduction of the load on the client, which, in turn, leads to a reduction in the cost of the system and reduce the requirements for hardware and software of clients. Logically, we can distinguish three levels of operations: the level of data representation, which, in essence, is a user interface and is responsible for presenting data to the user and the input of control commands from him; application level, which implements the basic logic of the application and at which the necessary information processing; level of data management, which provides data storage and access to them.

If it is not possible to identify a good architecture, the developer does not have clear agreements on how to structure the application. The lack of composite components leads to the appearance of code that has many dependencies. This type of code is difficult to understand. Adding new features becomes problematic, and it's not even clear where the new code should go.

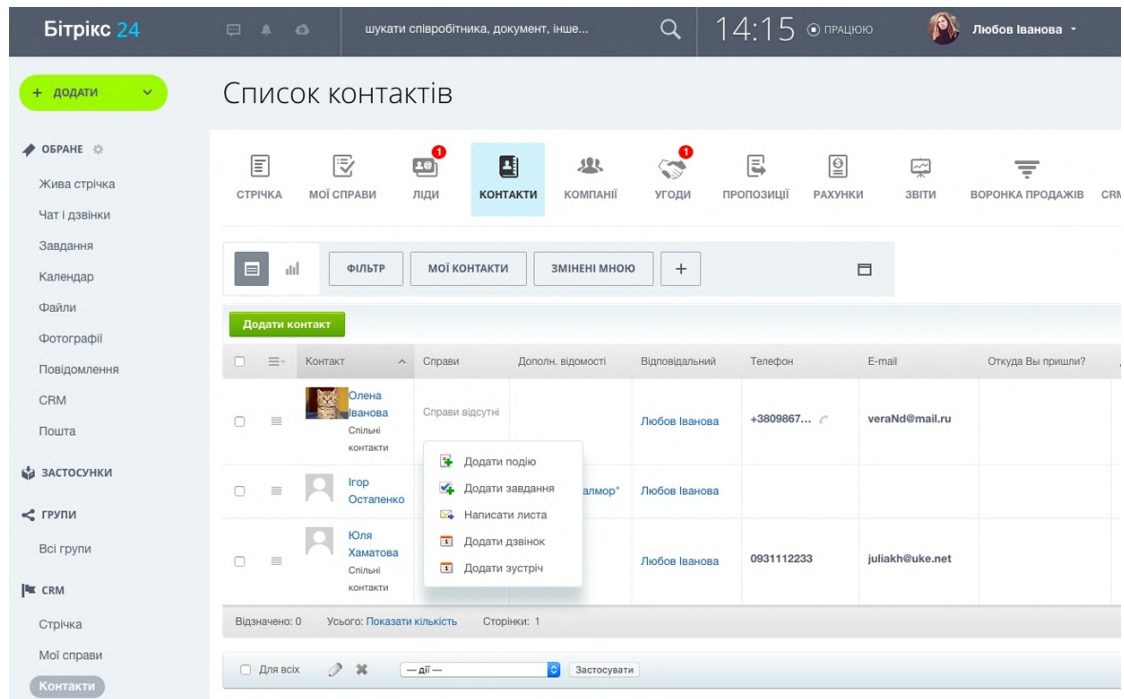
Some other potential problems are also common:

- The application has many variables, which complicates the knowledge of which widgets are rebuilt and when.
- It is unclear when certain variables may or may not be null because they are passed through multiple widgets.

All of these issues can significantly slow development and nullify the performance benefits that are common to Flutter.

In figure 6 shows a diagram of the architecture. Dotted horizontal lines define clear layers of





**Figure 5:** bitrix24.ua application interface.

the application. This comes down to the single responsibility principle: each component in the application must do only one.

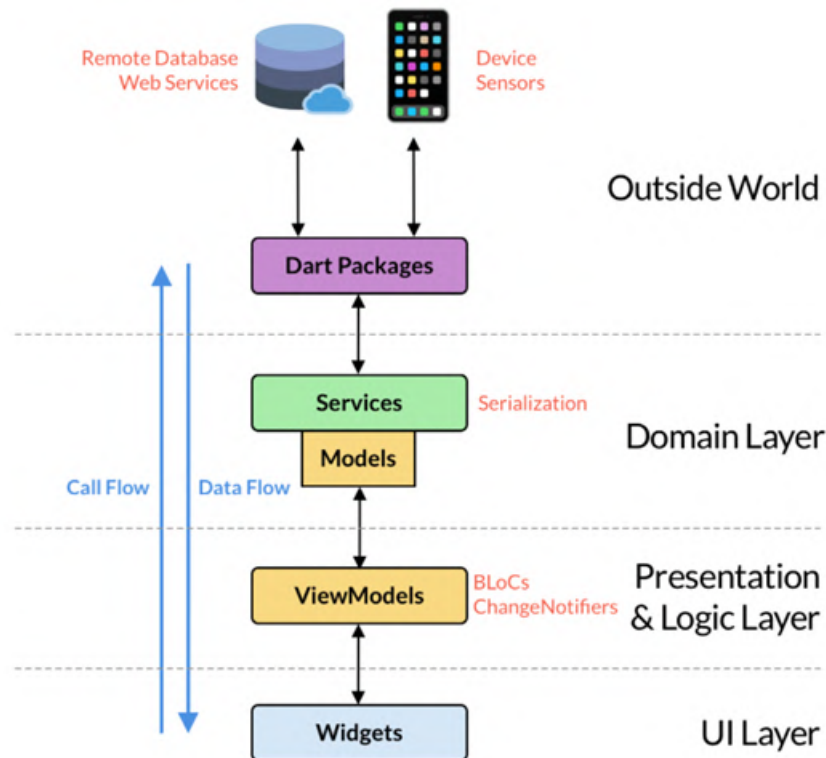
The following patterns can be used to separate business logic: MVC (Model View Controller); MVVM (Model View ViewModel); BLoC (Business Logic Component); MVP (Model View Presenter) [18].

This system is event-oriented, so the MVC pattern is not entirely appropriate. Because the View component will create events to which the Controller component must respond. And this, in turn, violates the concept of the pattern, because the View component should not affect the Controller. In addition, there are more appropriate patterns for such tasks.

MVVM consists of three components: models, presentation models, and presentations. This pattern is quite common in the design of mobile applications, which is quite justified because it is quite easy to use. MVVM is convenient to use instead of the classic MVC and the like in cases where the platform where the development is, there is a “data binding”. The MVVM architecture has a clear division of responsibilities. Functional connections between the user interface and the ViewModel are implemented through bindings, which are essentially rules such as “if the A button was pressed, the `onButtonAClick()` method of the ViewModel must be called”. Bandages can be written in code or defined declaratively [18].

BLoC (figure 7) is a pattern quite similar to popular solutions for mobile devices, such as MVP or MVVM.

It separates the level of presentation from the rules of business logic. This is a direct application of the declarative approach that Flutter emphasizes, ie  $UI = f(state)$ . BLoC is where events



**Figure 6:** Diagram showing the architecture for Flutter & Firebase.

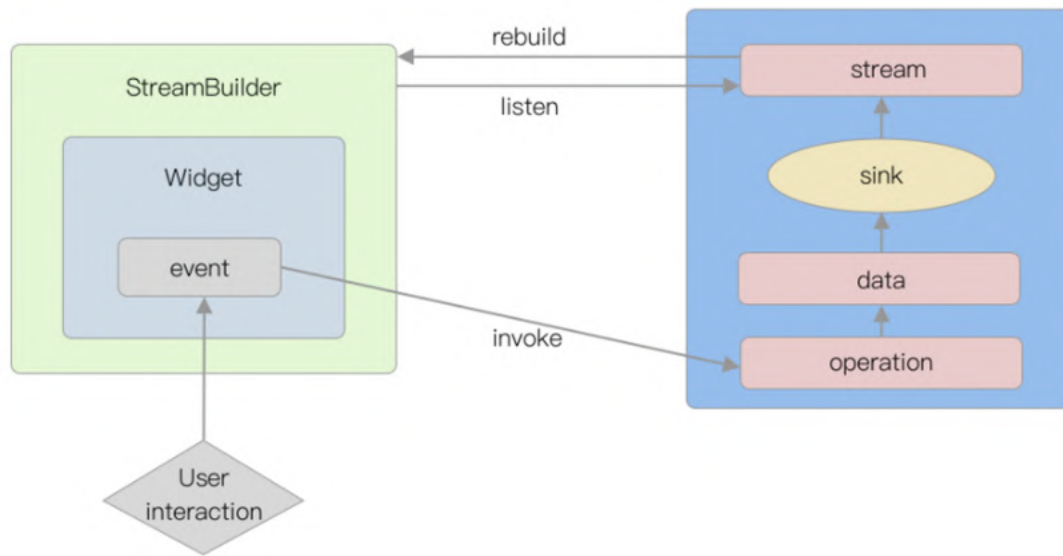
from the user interface go. Within this level, as a result of applying business rules to a given event, BLoC responds to a certain state, which then returns to the user interface. When the presentation level receives a new state, it restores its appearance following what the current state requires [18, 19].

MVP is a design template derived from MVC that separates the visual display and behavior of event processing into different classes, namely: View and Presenter. It is most often used for logically simple representations, but because most complex representations are logically complex, its use will also not be appropriate. In the case of simple representations, it can still be used, but mixing different patterns complicates the development and subsequent perception of the application code.

Therefore, to implement reactivity and meet the requirements of the mobile application, it will be advisable to use MVVM and BLoC patterns. The main advantage of BLoC over MVVM is that business logic can be completely separated from view, which is generally the best way to do something. As modern software development requires more and more changes to the user interface (as these are different screen sizes, densities, platforms, etc.), detaching the interface side from the models is a fantastic feature for reusing code.

Thus, the most flexible mobile application architecture was analyzed and selected.





**Figure 7:** BLoC pattern.

## 4.2. The rationale for the choice of tools

In the world of cross-platforms, all frameworks are approximately the same in structure. At the heart of everything - the target platform (iOS, Android), for which development is underway, and a layer of abstraction, which promises to do quickly, cheaply, and beautifully, and between them the bridge that connects the two entities.

This architecture has typical problems. They are partly because the layer of abstraction, relatively speaking, tries to sit on two chairs. Hence the difficulties: on each platform, the application is displayed and behaves differently. Sometimes this is not a problem, but there are times when the unique design and behavior on both platforms is an added business value [20].

In terms of requirements, one of the highest priorities is speed. Also, taking into account the experience of knowing these languages, the impression of developing mobile applications on these frameworks [21], and the availability of prototypes, Flutter was chosen.

Dart is an object-oriented and customer-optimized language for fast applications on any platform. The strengths of this language are:

- Improving the ease of development. JavaScript has a wide entry threshold. Dart should not complicate anything, but on the contrary, if possible to make the threshold easier (due to a clearer and more concise syntax), as well as to preserve the incorruptible and well-established essence of JavaScript.
- Increase productivity. Speed is one of the main problems of all modern technologies.
- Friendly to code editors and additional developer tools.
- Security.

The Firebase platform was chosen to work with the database. The main reason for choosing Firebase is the availability of the Firebase Realtime authentication service and database on

the platform in the cloud. The authentication service allows the user to link their accounts to the application, and accordingly – link any of their data. This service supports all standard authentication features. Each authorized user is provided with a unique Firebase user ID, which is used when integrating other Firebase services [22].

Work with the Firebase Realtime database is carried out in real-time without the need to make GET requests to the server. The data is stored in JSON format and synchronized with each connected client [22].

Finally, it is worth mentioning Firebase Storage. Firebase Storage is a service that supports downloading files from both the project and the project. It was decided to use this service in the application for reliability, convenience, and security. All uploaded photos will be stored in Firebase Storage.

### **4.3. Designing an object-oriented structure of the system**

Patterns need to be carefully selected and adjusted to the system. This is necessary for the implementation of effective solutions and their justified use. Next, will consider the patterns used in the design of the mobile application.

This mobile application is event-oriented. The components of the application interface that interact with the user are built on the BLoC pattern.

Events and actions are the input to the BLoC architecture. They are usually created in response to user interaction with the interface, such as button presses or life cycle events such as page loading. States are the results of the BLoC architecture. Moreover, they are part of the overall state of your application. Interface components can receive status notifications and redraw parts based on the current status.

BLoC (a component of business logic) converts a stream of input events into a stream of output states. BLoC is like a “brain” that receives information, processes it, and responds.

Stream is a sequence of asynchronous data. The user interface and BLoC listen to this stream and respond to any changes.

This application uses 13 BloCs:

1. AuthenticationBloc – the main block responsible for the definition of user authentication. Checks whether the user is logged in to his account, and if this is his first time, then send to the page to fill out the profile.
2. ChatBloc – using Stream, gets a list of chats of the current user.
3. GradebookBloc – using Stream, gets a log of grades depending on load options, such as student ID, group.
4. GroupBloc – to add a new group, also responsible for field validation.
5. GroupListBloc – using Stream, gets a list of groups depending on the load options.
6. LessonBloc – to add a new lesson with the choice of teacher, group, location, as well as fields such as homework, lesson topic, or regular class, or not blocked.
7. LessonListBloc – using Stream, gets a list of lessons depending on the load options.
8. LocationBloc – to add a new location, also responsible for field validation.
9. LoginBloc – is responsible for the user’s login. Responds to field changes using events (EmailChanged, PasswordChanged, LoginWithCredentialsPressed). Performs field validation.

10. MessagingBloc – is responsible for streaming messages, sending them. SendMessageEvent, MessageStreamEvent events.
11. ProfileBloc – is responsible for saving user profile information. Responds to field changes and validates them. Events NameChanged, BirthDateChanged, PhotoChanged, Submitted.
12. SignUpBloc – is responsible for user registration in the application, as well as field validation.
13. StudentListBloc – get a list of students of a certain group, or students who are not yet assigned to a group.

For all this to work, BLoC must be available. There are several ways to make it available. For example, through the global Singleton. This method is not recommended because Dart does not have a class destructor, so it will not be possible to free resources properly. Therefore, a separate instance of BLoC is created in the application. Initialization takes place inside the StatefulWidget.

The Repository architectural pattern was also used in this mobile application. Dart serves as an abstraction between the client code and the data provider, so as a developer working on features, you don't need to know where the data comes from. It can come from an API provider or a local database. So it's a good practice to use the Repository template.

The combined BLoC and Repository patterns work well together (figure 8).

The project uses 7 classes Repository (ChatRepository, GradebookRepository, GroupRepository, LessonRepository, LocationRepository, MessagingRepository, UserRepository).

#### 4.4. Development of data storage structure

The data storage structure plays an important role in project development. Namely, Cloud Firestore was chosen for storage. It is a flexible, scalable NoSQL cloud database for storing and synchronizing data for the client and server-side development. Cloud Firestore also offers seamless integration with other Firebase and Google Cloud products, including cloud features. And Storage for Firebase is a powerful, simple, and cost-effective object storage service. Firebase cloud storage SDKs add security to Google for uploading and downloading files for Firebase applications, regardless of network quality.

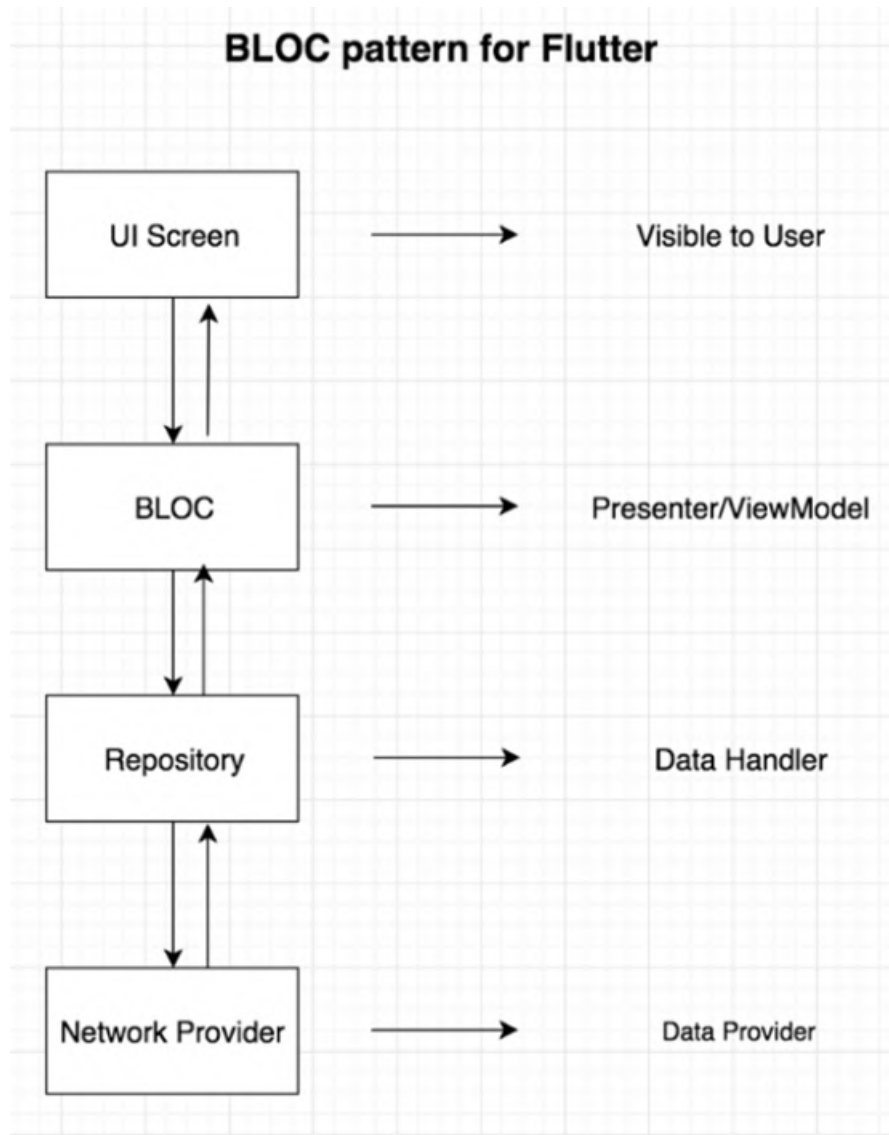
The following tables are required to implement the project database: messages, chats, users, gradebooks, gradebookRecords, groups, lessons, locations, cities.

Using Firebase, you can create subcollections. For example, the gradebooks collection, which is responsible for grades and attendance, contains a subset of records for each student.

#### 4.5. Design and implementation of algorithms for system operation

The first algorithm encountered by the user in the application is to determine whether the user is logged in. Then the role of the user is determined, whether it is a teacher, a student, or a manager. Depending on the role, the corresponding tab opens. In figure 10 shows the tab algorithm for the teacher.

If the role was defined as a student, then the first tab is displayed – GroupInfoPage. In figure 11 shows a diagram of student. For the student, the second tab is ClassmatesPage, where the student can view his profile information and a list of classmates. The second tab for the

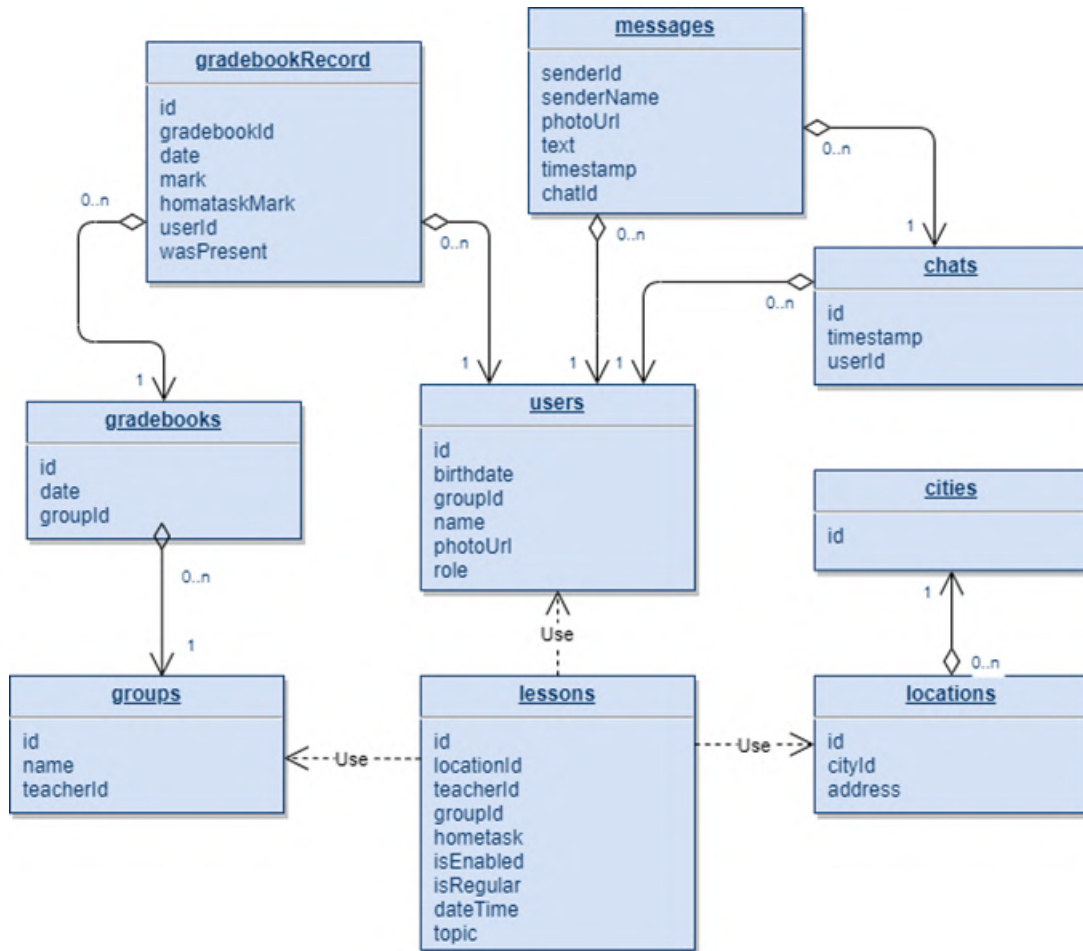


**Figure 8:** Using BLoC and Repository.

teacher is GroupLayoutPage with a list of groups and the ability to add new groups, locations, and lessons (figure 12).

#### 4.6. The procedure for installing and administering the system

The Flutter architecture differs from other software frameworks (React, Apache Cordova) in that it does not use HTML, CSS, and Javascript to build the interface, respectively, and the built-in WebView engine. It uses its engine for rendering. Flutter uses only one Dart programming language. Flutter is an innovative technology for developing cross-platform mobile applications



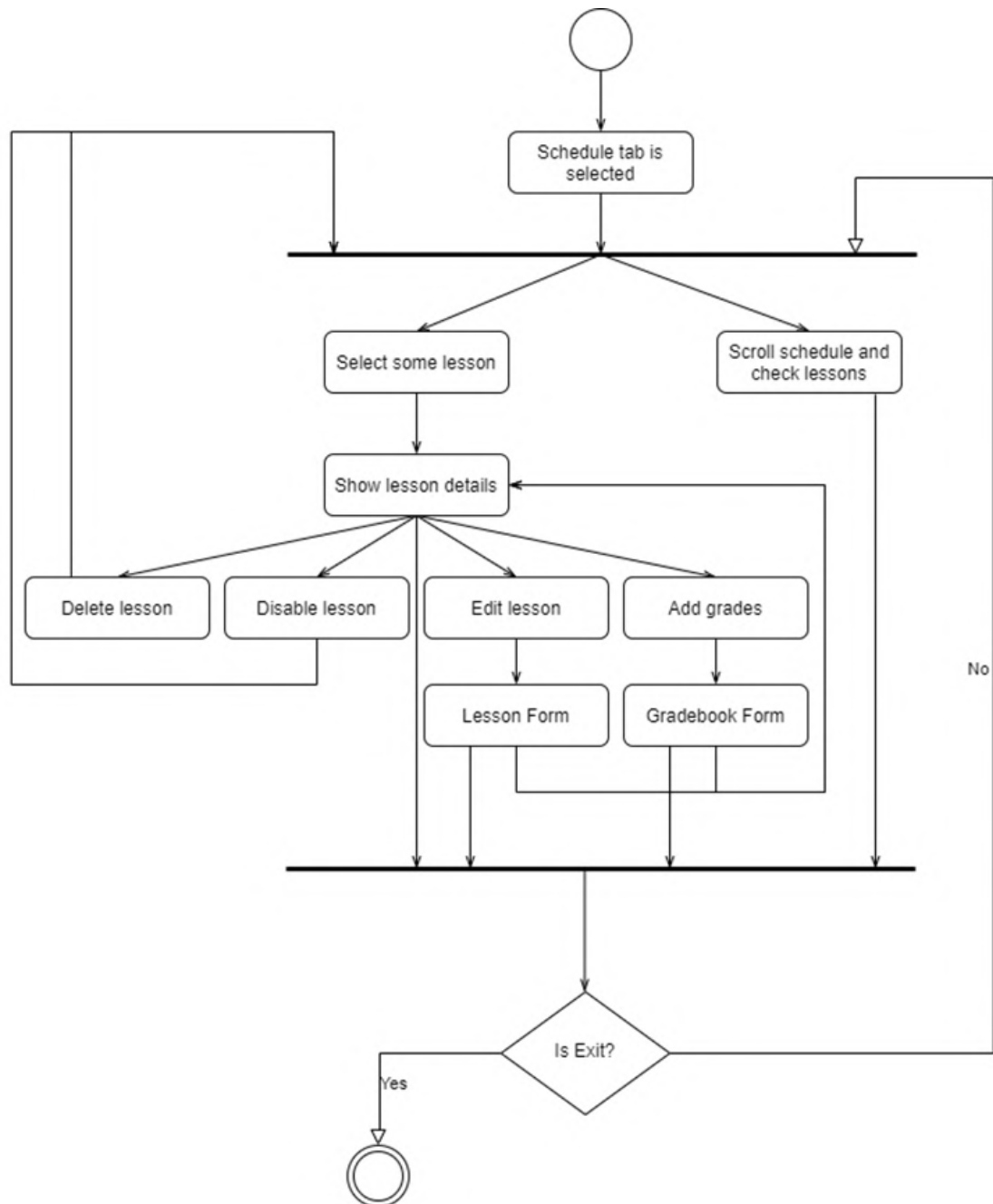
**Figure 9:** Scheme of the database.

for Android and iOS. Its advantage is the deployment of both Android and iOS at the same time using a single code base. Development is faster without losing productivity.

The chart is built using the online resource app.creately. In the diagram we see that it is divided into two component components, namely:

- Smartphone;
- Firebase Server is a serverless structure that allows you to automatically run server code in response to events caused by Firebase functions and HTTPS requests.

To exchange data between the client (we are considering the application on the smartphone) and the server (webserver) is mainly used HTTP protocol or its extension HTTPS for secure data transfer.

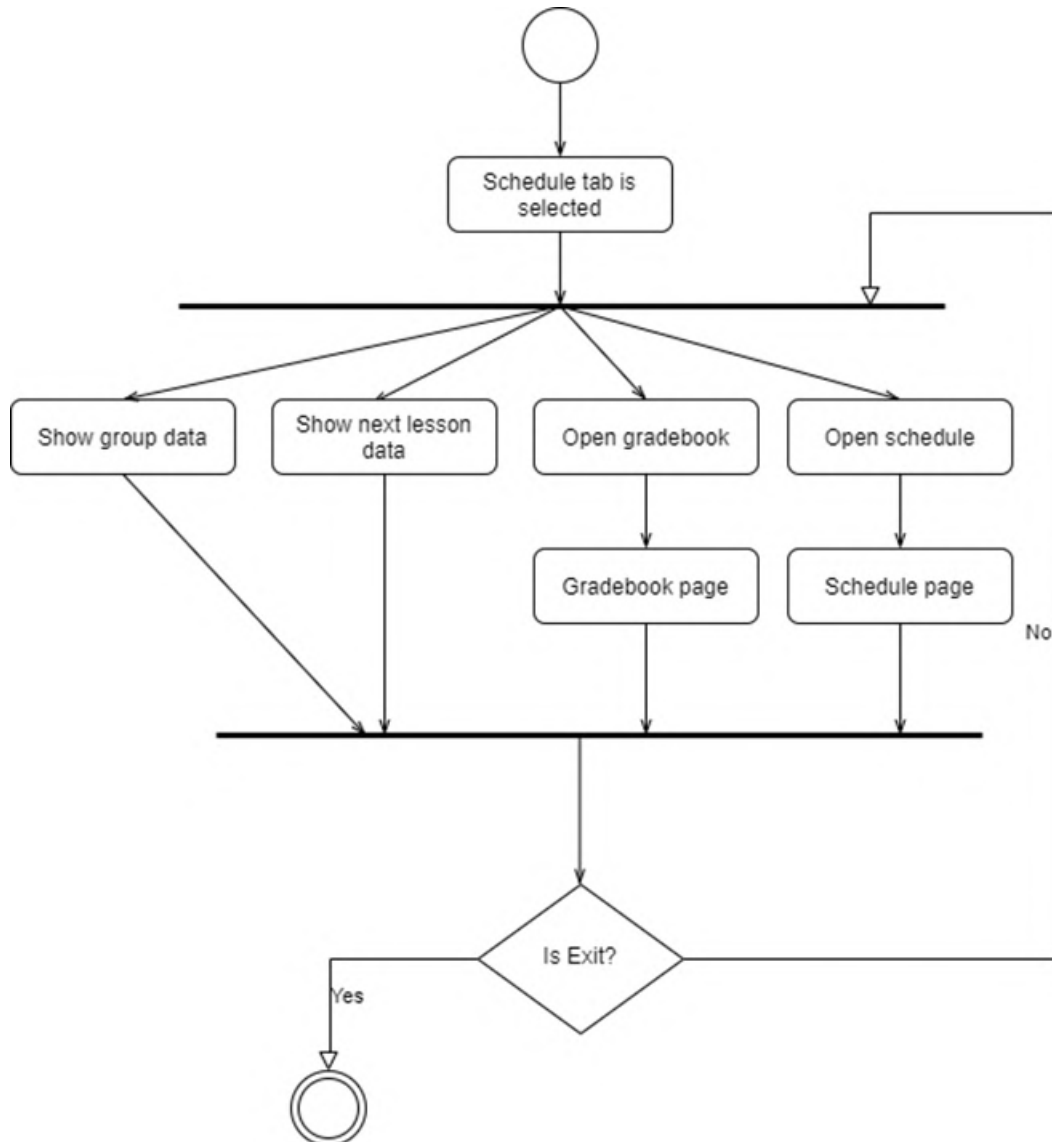


**Figure 10:** Diagram of teacher activity for the Schedule tab.

#### 4.7. Interface and procedure for working with the software package

On the desktop, the first thing the user sees is the application icon. It should be notable, interesting and include universal symbols that will be equally perceived by people with different



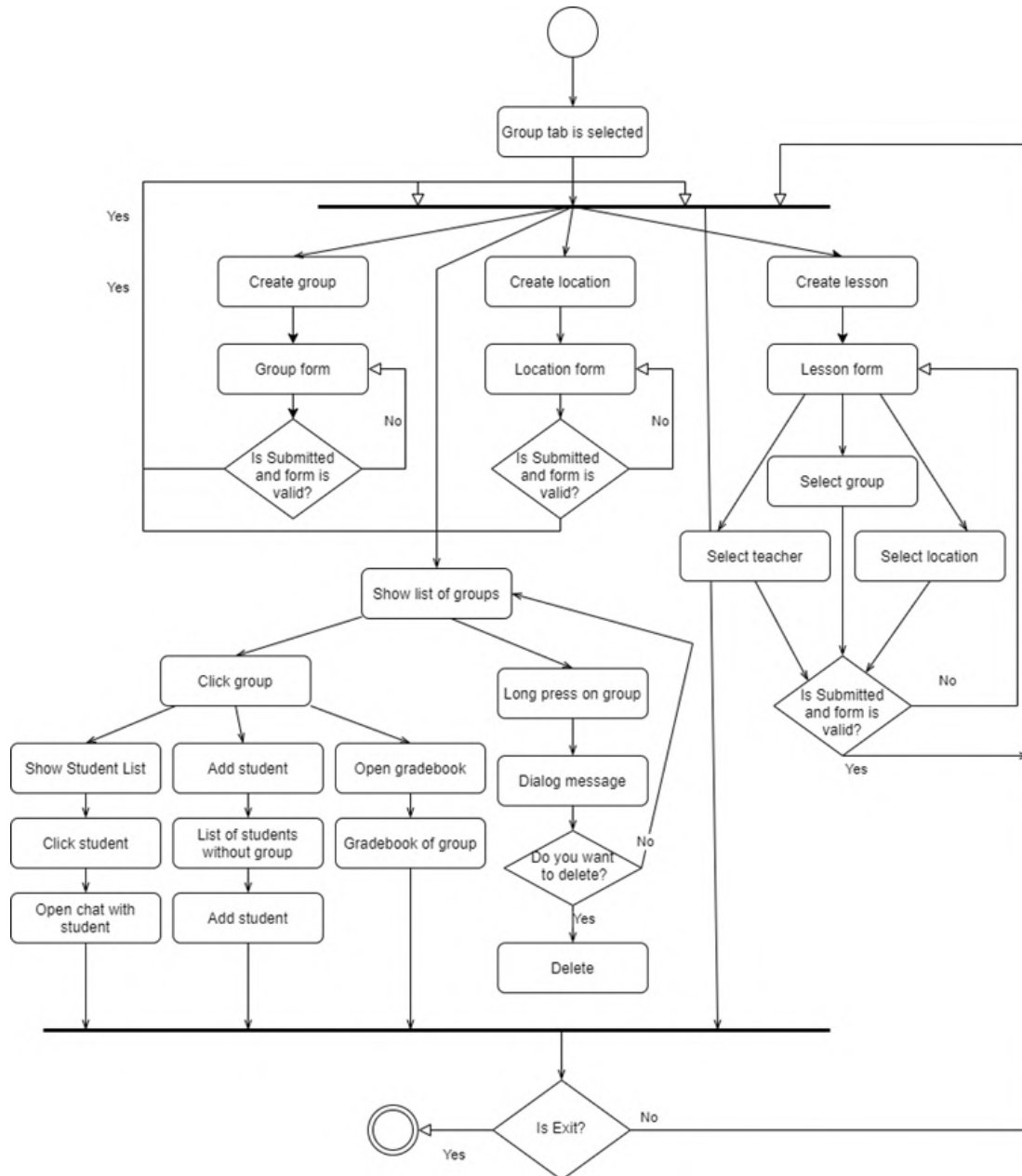


**Figure 11:** Student activity chart for the Schedule tab.

religious views. The style and logo were taken from the school Uprise Center, for which this product is developed.

The next thing that catches the user's eye is the download screen. It should also be stylistic and should be concise. It does not make sense to add a lot of text to it, because the application loads quickly and the user will not be able to read that information. Therefore, the design and text should be minimal.

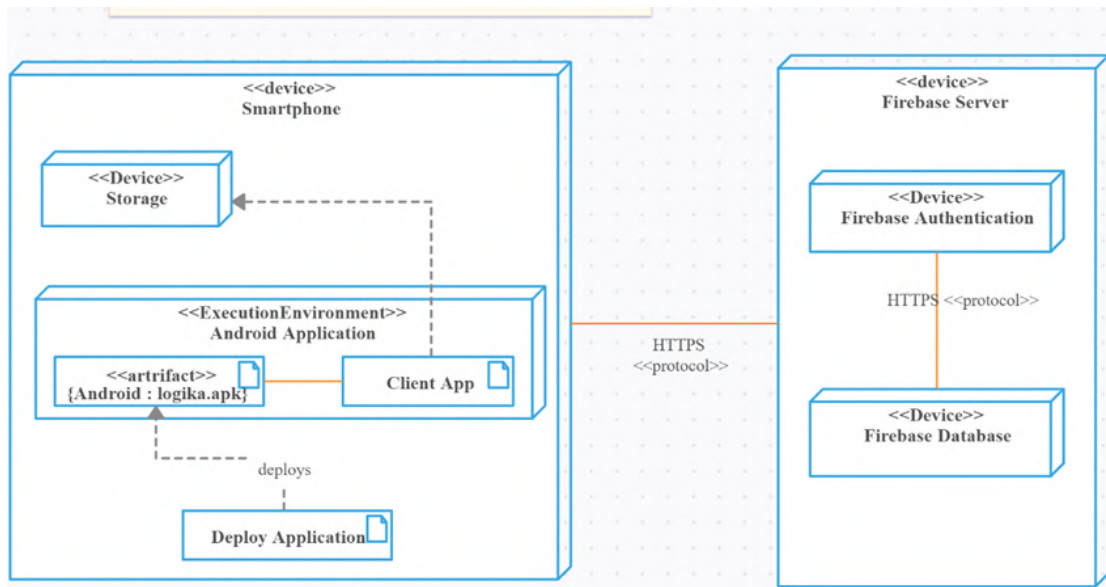
The unauthorized user is the first to see the authorization page. If the user is new, he can click on the text in the form and go to the registration page. When the fields are filled, the button becomes active and changes color.



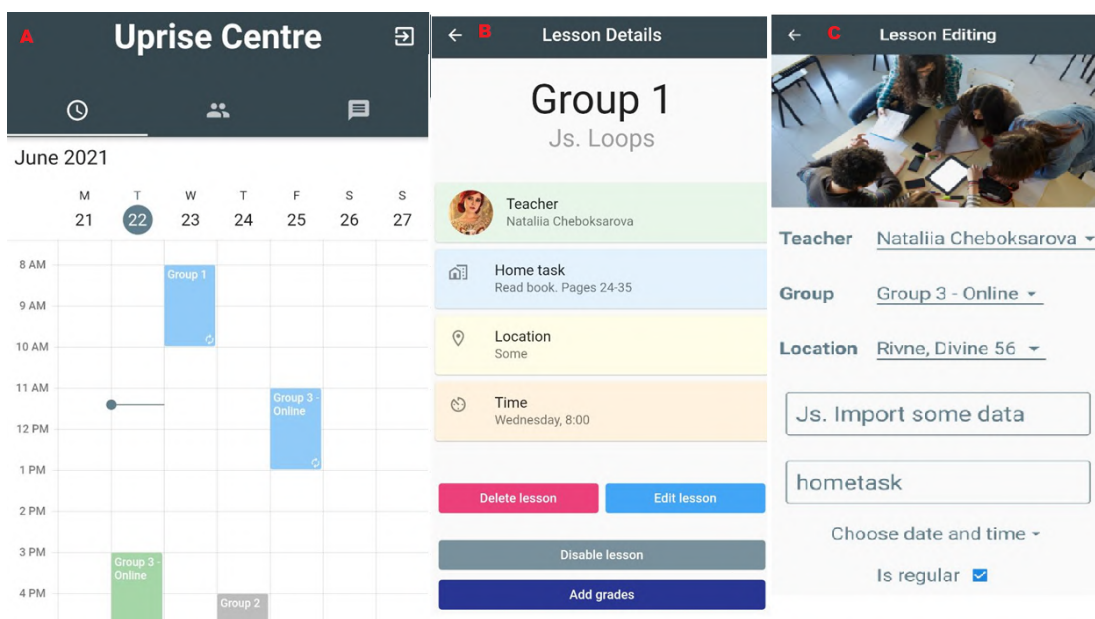
**Figure 12:** Diagram of teacher activity for the second tab.

To begin with, consider the type of application for the teacher. On the first page, we are greeted by the teacher's schedule (figure 14, A). Figure 15 shows the lessons in the schedule. Lessons differ in color because they have differences.

Blue classes are regular (held every week, `isRegular = true`), and active (`isEnabled = true`). This activity is displayed in the calendar each week on the appropriate day from the beginning



**Figure 13:** Deployment diagram.

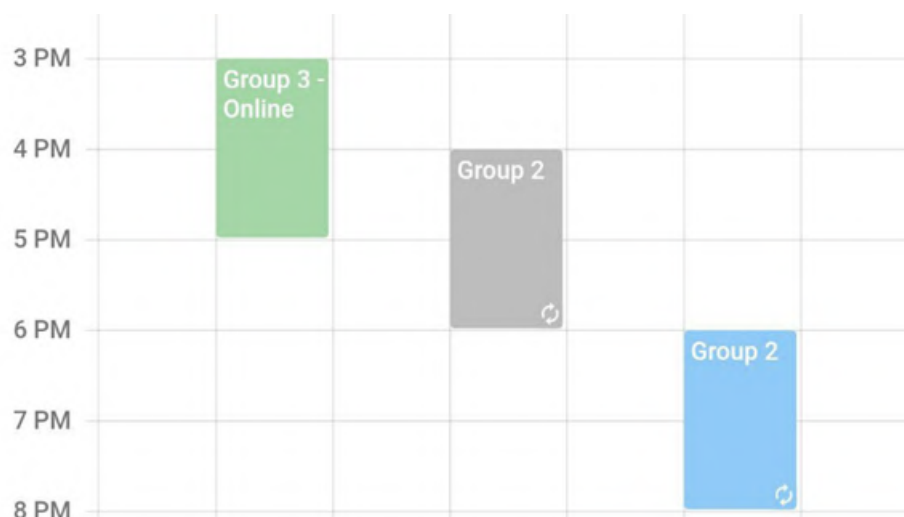


**Figure 14:** Teacher's schedule page.

of this lesson.

Green means that the lesson is single (ie will take place once, `isRegular = false`), active (`isEnabled = true`). This lesson is displayed in the calendar only once at a specified time.

Gray is a blocked activity (`isEnabled = false`). Whether it's a regular class or not, we can block



**Figure 15:** Display lessons in the schedule.

classes. That is, it will no longer appear in the student, but it can be activated at any time. This convenience helps to avoid deleting the lesson.

When you click on the lesson, detailed information about the lesson opens, and buttons for actions on the lesson are displayed (figure 14, B). For example, we can delete a lesson or edit it. Also, block the lesson and add grades to it.

The form for editing a lesson is the same as when creating a lesson. The difference is that it is already filled with data that is about this lesson. So that the user does not have to rewrite everything again, and he just changed the field he needs. The lesson editing page is shown in figure 14, C.

To create an entry in the journal, you must click on the “Add grades” button when editing the lesson. Then a form will open with a list of all students, where in front of each will be a tick that can be clicked and turned into a cross, which means whether the student was present. Also, next to each name you can put grades for the lesson and homework. Grades are given in a five-point system. By default, the student was present at the lesson, and the grades will be recorded as 0. The peculiarity of a school is that the grade for the lesson and homework is mandatory. Therefore, 0 is considered a full grade for poor performance in class. The completed and blank journal entry form is shown in figure 16. The second tab of the teacher is the tab of groups, as well as adding new locations, groups, and lessons.

When you click on the buttons Create group, Create lesson, Create location, the corresponding form of creation opens. Validation is calculated for each field. And if the form is not valid, the button is not active.

For such fields as City, Group, Location, Teacher the drop-down list from the received data is provided.

The last tab is chats, it is common to all users. It shows a list of chats. You can see the name and photo of the user you were talking to, as well as the last chat message and how long ago this message was sent.

Group Adding			
Karambolka	✓	mark ▾	h/w mark ▾
Dima Sych	✓	mark ▾	h/w mark ▾
Petro Zima	✓	mark ▾	h/w mark ▾
Svitlana Scott	✓	mark ▾	h/w mark ▾
Kolos Oksana	✓	mark ▾	h/w mark ▾

\* all unset marks will be considered zero

Save

Group Adding			
Karambolka	✗	mark ▾	h/w mark ▾
Dima Sych	✓	2 ▾	5 ▾
Petro Zima	✗	mark ▾	h/w mark ▾
Svitlana Scott	✓	2 ▾	1 ▾
Kolos Oksana	✓	5 ▾	5 ▾

\* all unset marks will be considered zero

Save

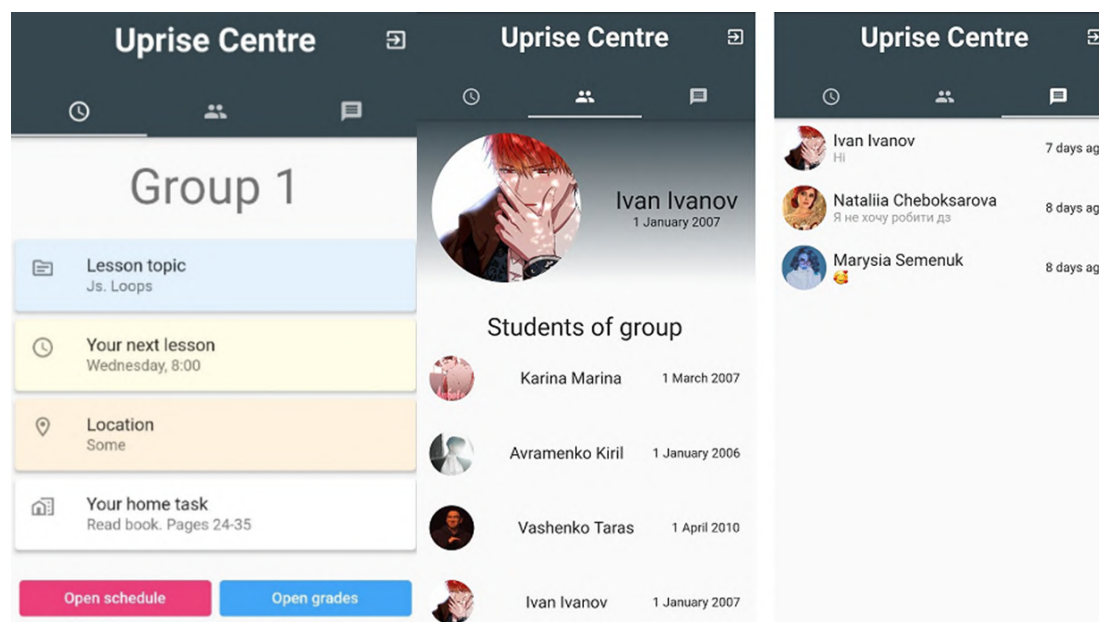
**Figure 16:** Empty and filled form of assessments and attendance at the lesson.

When you click on the chat it opens and all messages are displayed, showing who sent them and when. Also below is a text entry field, as well as a button to add a photo. When opened, the chat is initially scrolled down and shows the last message. You can send both short and long messages, photos, emoticons.

When you long-click on a group in the group list, or a chat in the chat list, or a student in the corresponding list, a dialog box will pop up, asking if you want to delete the item. When you select “no”, the action is canceled. If you agree, the item will be deleted and the list will be updated, so the deleted item will no longer be displayed.

For the student, the appearance of the first two tabs will be different. For example, the first tab will show the name of the group and detailed information about the next lesson (figure 17). The next lesson is based on single and regular lessons but ignores the lessons that are blocked. There are also buttons to show grades and schedules, which work in the same way as the teacher. The difference is that only this student’s grades and schedule will be shown.

On the second tab, the student is waiting for his profile information, which he filled out after registration, as well as a list of all the children in the group to which he is attached. Clicking on any student will open a chat with him. You can also open chats, as in the teacher, going to the last tab (figure 17).



**Figure 17:** All tab of the student.

## 5. Conclusions

In this work, a software product was developed – a CRM system with a mobile application for a school. In the course of the work, the processes of modeling, building architecture, and implementation of web application functionality were described. This CRM system is implemented in the educational process of a school and undergoes initial testing. Further development and support of the developed software product are planned. Thanks to a properly designed architecture, these processes will take place without undue delay in adapting the structure of the application.

## References

- [1] Terrasoft, Shcho take CRM-systema ta yak vona pratsiuie? (What is a CRM system and how does it work?), 2022. URL: <https://www.terrasoft.ua/page/definition-crm>.
- [2] O. Kuzminska, M. Mazorchuk, N. Morze, O. Kobylin, Attitude to the digital learning environment in Ukrainian universities, CEUR Workshop Proceedings 2393 (2019) 53–67.
- [3] O. P. Pinchuk, O. M. Sokolyuk, O. Y. Burov, M. P. Shyshkina, Digital transformation of learning environment: Aspect of cognitive activity of students, CEUR Workshop Proceedings 2433 (2019) 90–101.
- [4] O. H. Kolgatin, L. S. Kolgatina, N. S. Ponomareva, E. O. Shmeltser, A. D. Uchitel, Systematicity of students' independent work in cloud learning environment of the course "Educational Electronic Resources for Primary School" for the future teachers of primary schools, in: S. Semerikov, V. Osadchyi, O. Kuzminska (Eds.), Proceedings of the Sym-



- posium on Advances in Educational Technology, AET 2020, University of Educational Management, SciTePress, Kyiv, 2022.
- [5] L. V. Vasylieva, D. Y. Mikhieienko, I. A. Getman, M. V. Kormer, Opportunities and ways of using laboratory equipment in a distance learning environment, in: S. Semerikov, V. Osadchyi, O. Kuzminska (Eds.), Proceedings of the Symposium on Advances in Educational Technology, AET 2020, University of Educational Management, SciTePress, Kyiv, 2022.
  - [6] N. B. Pasko, O. B. Viunenko, S. V. Agadzhanova, K. H. Ahadzhanov-Honsales, Using intelligent agent-managers to build personal learning environments in the e-learning system, in: S. Semerikov, V. Osadchyi, O. Kuzminska (Eds.), Proceedings of the Symposium on Advances in Educational Technology, AET 2020, University of Educational Management, SciTePress, Kyiv, 2022.
  - [7] A. D. Uchitel, I. V. Batsurovska, N. A. Dotsenko, O. A. Gorbenko, N. I. Kim, Implementation of future agricultural engineers' training technology in the informational and educational environment, CEUR Workshop Proceedings 2879 (2020) 233–246.
  - [8] R. Munk, D. Marchant, B. Vinter, Cloud enabling educational platforms with corc, CEUR Workshop Proceedings 2879 (2020) 438–457.
  - [9] K. V. Vlasenko, S. V. Volkov, I. V. Lovianova, O. O. Chumak, I. V. Sitak, D. Y. Bobyliev, Management of online platform development and support process, CEUR Workshop Proceedings 2879 (2020) 333–345.
  - [10] L. Y. Sultanova, O. P. Tsiuniak, L. O. Milto, M. O. Zheludenko, L. M. Lyktei, L. M. Petrenko, A. D. Uchitel, The potential of Google Classroom web service for lecturers of higher educational establishments under pandemic conditions, CEUR Workshop Proceedings 2879 (2020) 346–365.
  - [11] M. P. Shyshkina, Service models of the cloud-based learning environment of the educational institution, CEUR Workshop Proceedings 2168 (2017) 1–6.
  - [12] M. V. Popel, M. P. Shyshkina, The areas of educational studies of the cloud-based learning systems, CEUR Workshop Proceedings 2433 (2019) 159–172.
  - [13] N. V. Soroko, L. A. Mykhailenko, O. G. Rokoman, V. I. Zaselskiy, Educational electronic platforms for STEAM-oriented learning environment at general education school, CEUR Workshop Proceedings 2643 (2020) 462–473.
  - [14] K. Osadcha, V. Osadchyi, S. Semerikov, H. Chemerys, A. Chorna, The review of the adaptive learning systems for the formation of individual educational trajectory, CEUR Workshop Proceedings 2732 (2020) 547–558. URL: <http://ceur-ws.org/Vol-2732/20200547.pdf>.
  - [15] D. S. Antoniuk, T. A. Vakaliuk, V. V. Ievdokymov, A. V. Morozov, V. V. Kontsedailo, Integrating business simulations software into learning environment of technical university, Journal of Physics: Conference Series 1946 (2021) 012018. doi:10.1088/1742-6596/1946/1/012018.
  - [16] T. A. Vakaliuk, D. S. Antoniuk, V. N. Soloviev, The state of ICT implementation in institutions of general secondary education: A case of Ukraine, CEUR Workshop Proceedings 2643 (2020) 119–133.
  - [17] T. A. Vakaliuk, D. S. Antoniuk, O. O. Kalinichenko, The state of ICT implementation in Ukrainian general secondary education institutions in 2019 and 2020, in: S. Semerikov, V. Osadchyi, O. Kuzminska (Eds.), Proceedings of the Symposium on Advances in Educational Technology, AET 2020, University of Educational Management, SciTePress, Kyiv,

- 2022.
- [18] MVVM vs BLoC, 2018. URL: [https://www.reddit.com/r/FlutterDev/comments/9y8nxb/mvvm\\_vs\\_bloc/](https://www.reddit.com/r/FlutterDev/comments/9y8nxb/mvvm_vs_bloc/).
  - [19] Bloc State Management Library, 2022. URL: <https://bloclibrary.dev>.
  - [20] N. Sharma, React Native vs. Xamarin vs. Ionic vs. Flutter: Which is best for Cross-Platform Mobile App Development?, 2018. URL: <https://www.apptunix.com/blog/frameworks-cross-platform-mobile-app-development/>.
  - [21] Layouts in Flutter, 2021. URL: <https://flutter.dev/docs/development/ui/layout>.
  - [22] Firebase Documentation, 2021. URL: <https://firebase.google.com/docs>.