

NATIONAL RESEARCH UNIVERSITY
HIGHER SCHOOL OF ECONOMICS

Faculty of Computer Science
Bachelor's Programme "Data Science in Business Analytics"

**Software Project Report on the Topic:
App HSE Grades for students**

Submitted by the Student:

group #БПАД221, 2nd year of study Klevtsova Snezhana Vadimovna

Approved by the Project Supervisor:

Borevsky Andrey Olegovich

Assistant

Department of Data Analysis and Artificial Intelligence, HSE University

Contents

- Annotation** **3**

- Keywords** **5**

- 1 Introduction** **6**

- 2 Analyzing of analogues** **8**
 - 2.1 Review of analogues 8
 - 2.2 Analyzing of the HSE Grades app 10

- 3 Functionality of the app** **11**

- 4 Technical issues** **17**
 - 4.1 Mechanism of the app 17
 - 4.2 Security 18
 - 4.3 Technical issues 18

- 5 Conclusion** **20**

- References** **21**

Annotation

Nowadays a lot of enterprises try to implement vertical integration where all business processes occur within a single system owned by the company itself. This improves the level of data protection and makes it easier to use.

HSE has several apps and services used by students and employees. However, some functionality is not presented: there is no opportunity to look at cumulative grades, components of current grades (points for exercises in a task), submit a work and ask a question about it in one application. Consequently, using of external services is needed such as Google Sheets, Google Classroom, Yandex Disk etc. This causes great inconvenience for teachers and students due to the large amount of time spent searching and providing the necessary information. Moreover, the use of resources that do not belong to the HSE University may lead to data leakages and other problems associated with cyber fraud.

The *HSE Grades* app provides most of the information needed for and control of the educational process; has convenient functionality for detailed viewing of grades, work of teaching assistants and teachers. The security of using the application is achieved through the use of a classic client-server architecture using secure communication channels and a token system, as well as the use of a role model of process participants.

Аннотация

В настоящее время многие предприятия стараются перейти к стратегии вертикальной интеграции, где все бизнес-процессы происходят внутри единой системы, принадлежащей самой компании. Это позволяет улучшить степень защиты данных и упростить пользование ими.

НИУ ВШЭ имеет несколько приложений и сервисов, которыми пользуются сотрудники и студенты. Тем не менее, некоторый функционал в них не представлен: отсутствует возможность посмотреть кумулятивные оценки, компоненты текущих оценок (баллы за упражнения в задании), сдать работу и задать вопрос по ней в одном приложении. В связи с этим появляется необходимость пользоваться сторонними сервисами, такими как Google Sheets, Google Classroom, Yandex Disk и другие. Это вызывает большие неудобства для преподавателей и учащихся в связи с большими временными затратами на поиск и предоставление нужной информации. Кроме того, использование ресурсов, не принадлежащих НИУ ВШЭ, может повлечь утечки данных и другие проблемы, связанные с кибермошенничеством.

Приложение *HSE Grades* предоставляет основную часть информации, необходимой для образовательного процесса и контроля над ним; имеет удобный функционал для детального просмотра оценок, работы учебных ассистентов и преподавателей. Безопасность использования приложения достигается за счет использования классической клиент-серверной архитектуры с задействованием защищенных каналов связи и системы токенов, а также использованием ролевой модели участников процесса.

Keywords

- **Application (app)** — it is a computer program designed to carry out a specific task other than one relating to the operation of the computer itself, typically to be used by end-users;
- **Database** — is an organized collection of structured information, or data, typically stored electronically in a computer system;
- **SQL** — a programming language designed for accessing, modifying and extracting information from databases ([5]);
- **Qt** - is cross-platform software for creating graphical user interfaces as well as cross-platform applications that run on various software and hardware platforms ([7]).
- **Client-server architecture** — is architecture of a computer network in which many clients (remote processors) request and receive service from a centralized server (host computer). Client computers provide an interface to allow a computer user to request services of the server and to display the results the server returns. Servers wait for requests to arrive from clients and then respond to them.
- **SSL (Secure Sockets Layer)** — an encryption-based set of Internet security protocol which use HTTPS instead of HTTP.
- **HTTP (HyperText Transfer Protocol)** — an application layer protocol designed to transfer information between networked devices (usually between Client and Server) ([1]). **HTTPS** is HTTP which transfers data through channels defended by SSL.
- **Token** — is a virtual entity formed by server after successful client authentication and used for specific security task - determining HTTP session role for example - without re-quiring security critical information such as password. Besides, As a rule, in addition to the role model, tokens also contain information about the IP address of an authenticated user, which allows for additional protection against token substitution ([2]).
- **Three Tier architecture** — architectural software model in which application splitted to 3 logical layers: presentation tier (communication with a client - REST controllers, for example), application (business) tier (processing of requests, service) and data access tier (forward communication with a database) ([2]).

1 Introduction

Creating a closed ecosystem is important for any enterprise. The use of resources not controlled by the structure may lead to data leakage, doxing, various types of fraud, and further use of the obtained information against the company's employees and its customers. In addition, often the need to interact with other services, switching from the main one to others, can be extremely inconvenient for all users.

This problem also affects universities, including the Higher School of Economics. HSE has several sources of information used by staff and students, such as LMS HSE, SmartLMS, RUZ and HSE App X, with some functions repeated in different applications (see table). But the most important problem is the use of third-party services: for almost all subjects additional tables in Google Sheets are created to record grades and attendance, various cloud storages (such as Google and Yandex Disks) are used to store electronic materials for study.

Such issues exist:

1. Information about students' grades in different subjects is stored in different tables, that is, the student does not have the opportunity to view all his grades in one place. Consequently, more time is spent tracking attendance.
2. If the services fail, both teachers and students will not have access to the materials. In addition, in this case, the risk of data leakage about student performance increases.
3. Neither application supports viewing assessment components. By this it is meant that most often either only the final grade for the subject is shown, or the grade for the entire assignment is visible, but there is no information about the points received for sub-tasks of the assignments. This is very inconvenient for students and they have to ask teaching assistants and teachers about it.
4. Teachers and assistants have to spend their time and energy compiling tables and writing formulas.
5. It is difficult for study office to control the educational process since all information is in different places (tables).

Created app *HSE Grades* solves these problems. It provides interface which allows to see information about all subjects just switching filter settings. A student can see his performance in tables where there are such columns as "Subject", "Mark", "Time of submission", "Name of assignment" and all other necessary information, including cumulative grades. Not exiting the

app it is possible to view available assignments, submit work and ask for consultation if needed. From the side of teaching assistants, they can add tasks, view submitted works and grade them. Creating of grading tables (for sub tasks) is simple: all students are already in the table, it is only needed to define names of columns and put needed values. Teachers can control work of teaching assistants.

The app is operating on the base of client-sever architecture and stores data in MySQL database. Code is organized according to Three Tier architecture model which allow to control the process of data transmission on all stages (communication with a client, handling of a request by the app and getting/giving information to the database). Also *HSE Grades* provides multi threading so if there will be several users at one moment the app will work with them in parallel threads and not delay requests. Finally, the app is well-secured: it uses token-based authentication (to be discussed further), encryption of data via Envelope algorithm based on AES (Advanced Encryption Standard).

2 Analyzing of analogues

2.1 Review of analogues

As it was mentioned before, there are 4 main services which are used by HSE employees and students: LMS HSE, SmartLMS, RUZ and HSE App X. To develop an app which is suitable for educational process it is important to highlight features which are essential but not presented in existing variants. In this case it is not needed to look at RUZ in detail since it only provides schedule of lessons.

As it can be seen from the table [2.2](#), there are several problems which not allow to work efficiently.

1. Apps allow to see final grades for the courses but do not show marks for homeworks, quizzes etc.
2. SmartLMS has a big potential. However, since some information is filled by the teaching staff, a lot of SmartLMS course pages are "blank". Consequently, it is not possible to use only SmartLMS as a source of information.
3. Some apps have similar functions but are not synchronized provoking collisions. Therefore, it is needed to develop one app which will contain all needed features.
4. As stated in the first paragraph, intermediate estimates are not shown. This is largely due to the lack of functions that allow teachers to create tables for a detailed description of tasks. Consequently, teachers use services that provide this opportunity (for example, Google Sheets). Thereafter, it is very important to solve this problem to restore the ecosystem of the university.
5. For students it is essential that the app does not only show grades, but allow to view current assignments, find out deadlines, submit works and ask questions. Therefore, it is crucial to provide functional interface.

All these issues must be taken into account in the application.

Table 2.1: Table 1. Comparison of Smart LMS, LMS HSE and HSE App X in terms of different characteristics.

Issue	Smart LMS	LMS HSE	HSE App X
Opportunity to see final grades	Yes; however, sometimes teachers do not fill SmartLMS pages	Yes	Yes
Opportunity to see components of grades	Yes; but it also depends on whether teachers filled pages	No	No
Opportunity to see current assignments	Yes; but it also depends on whether teachers filled pages	No	No
Opportunity to submit works	Yes; but it also depends on whether teachers filled pages	No	No
Opportunity to check the schedule of lessons	No	No	Yes
Using of non-HSE services	Yes; full description of grades is presented in Google or Yandex Sheets	No (since do not provide detailed description of grades)	No (since do not provide detailed description of grades)
Teaching assistants can grade works	Yes	No (only Teaching Office)	No (only Teaching Office)
Other services are provided (scholarships, useful links)	No, only information about courses	Yes, it contains services connected with educational process	Partially: it provides information about canteens, library
Problems detected by students	<ul style="list-style-type: none"> • When there is a big inflow of student requests SmartLMS stops its work; • sometimes there are difficulties with uploading tasks; • teaching staff does not fill SmartLMS pages because of some inconveniences - > not full information is presented. 	<ul style="list-style-type: none"> • Information in LMS HSE is added and edited too rare; • It contains only final grades. 	<ul style="list-style-type: none"> • One of the main purpose of this app is to provide quick access to the schedule; however, frequently schedule in HSE App X and RUZ (more official) differ; • In this app some information about other students can be seen by any student.

2.2 Analyzing of the HSE Grades app

The app HSE Grades has the following characteristics:

Table 2.2: Table 1. Characteristics of the HSE Grades app.

Issue	HSE Grades	Comment
Opportunity to see final grades	Yes	There is a special window "Results" which shows intermediate and final grades; also there is opportunity to "predict" grades via entering other values
Opportunity to see components of grades	Yes	Grades for homeworks, quizzes, tests etc. are shown; also evaluation of subtasks can be seen in the table (only student grades)
Opportunity to see current assignments	Yes	They can be seen in the "Assignments" window
Opportunity to submit works	Yes	It can be done by clicking on the needed assignment and fill a submission form
Opportunity to check the schedule of lessons	No	Probably in the future versions
Using of non-HSE services	No	The main purpose of the app was to escape this
Teaching assistants can grade works	Yes	They can both add assignments and grade submitted works (also creating grading table for subtasks)
Other services are provided (scholarships, useful links)	No	In the future updates

To sum up, the app *HSE Grades* has the most important functionality. In the next section all possible actions are described in details.

3 Functionality of the app

When the user only open the app he sees the following window (3.1):

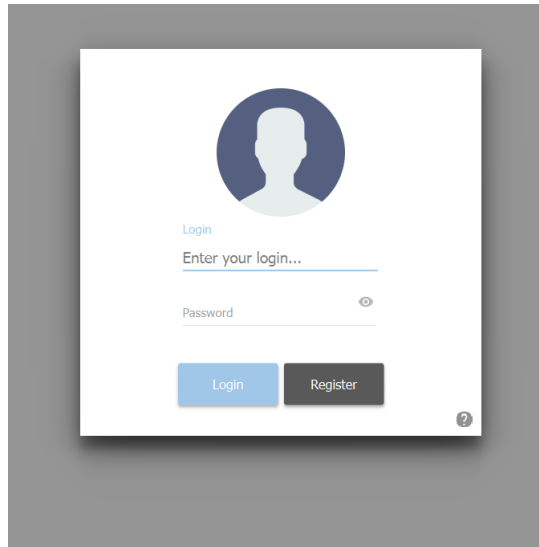


Figure 3.1: Entrance window.

Here a user enters his login and password. If he is new, he can register via clicking on the "Register" button. He will see the registration window (3.2) where it is needed to enter all important information about him: Now suppose three cases: if he is a student, a teaching assistant

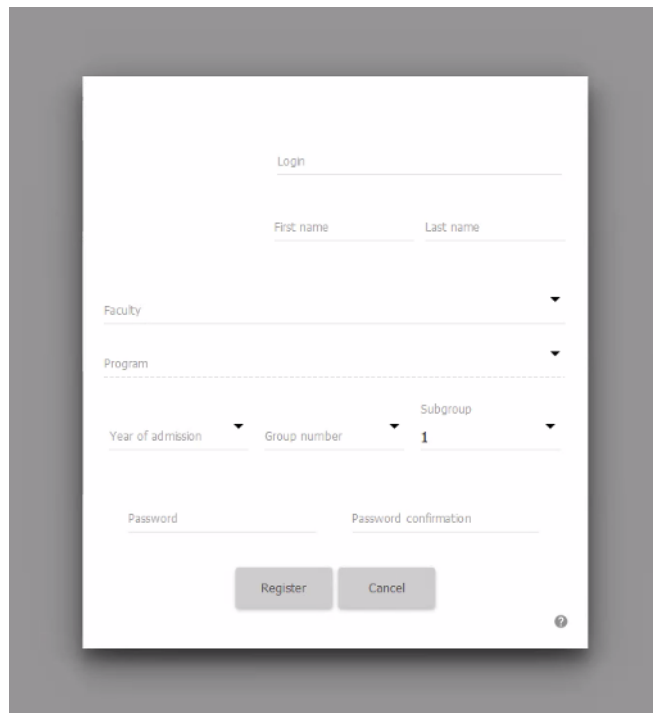


Figure 3.2: Registration window.

or a teacher.

1. Student.

When he enters its credentials the following window (3.3) can be seen: On the right there

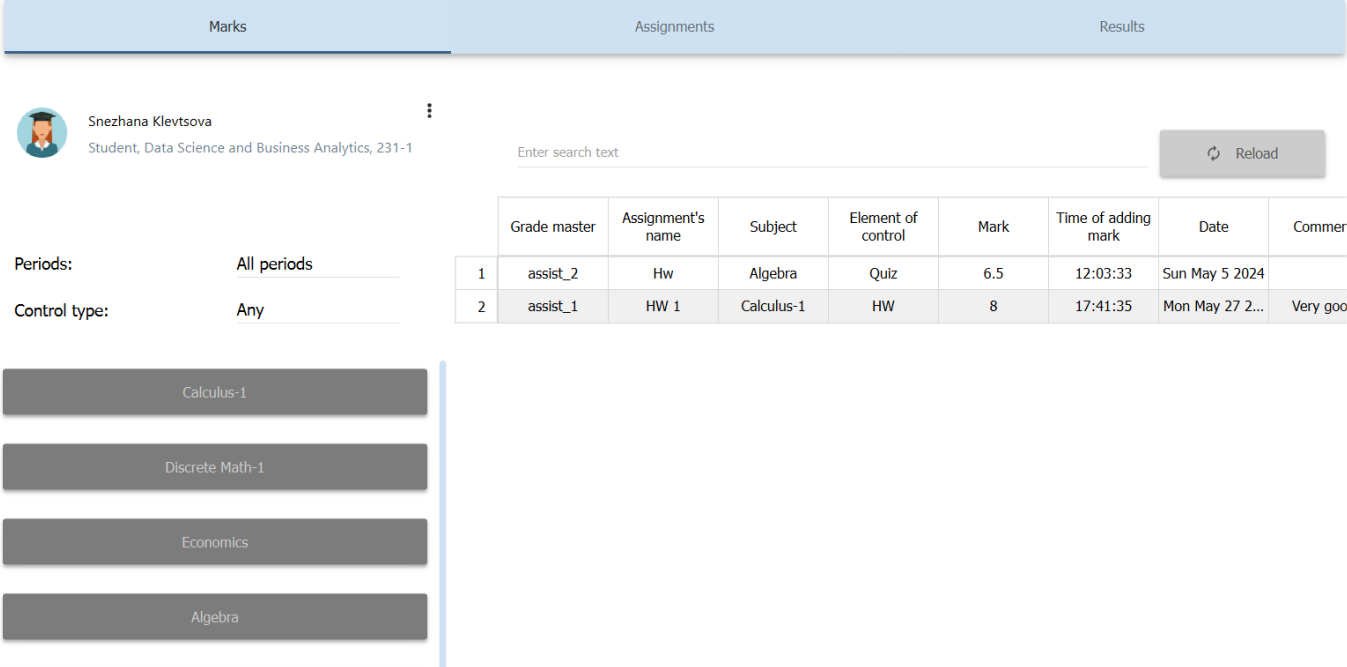


Figure 3.3: Student grades window.

is a table with grades. Here a student can see his mark, name of assignment, subject etc. On the left there is a filter by subject, element of control and time period. On the left top corner there is a profile card (3.4) :

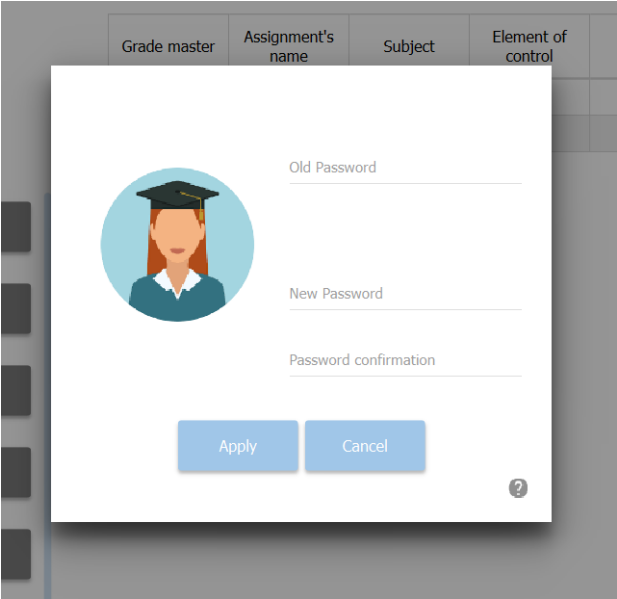


Figure 3.4: Student profile card.

Here student can change his password and avatar if needed.

Further he can look for current assignments (3.5): On the left there is a list of available

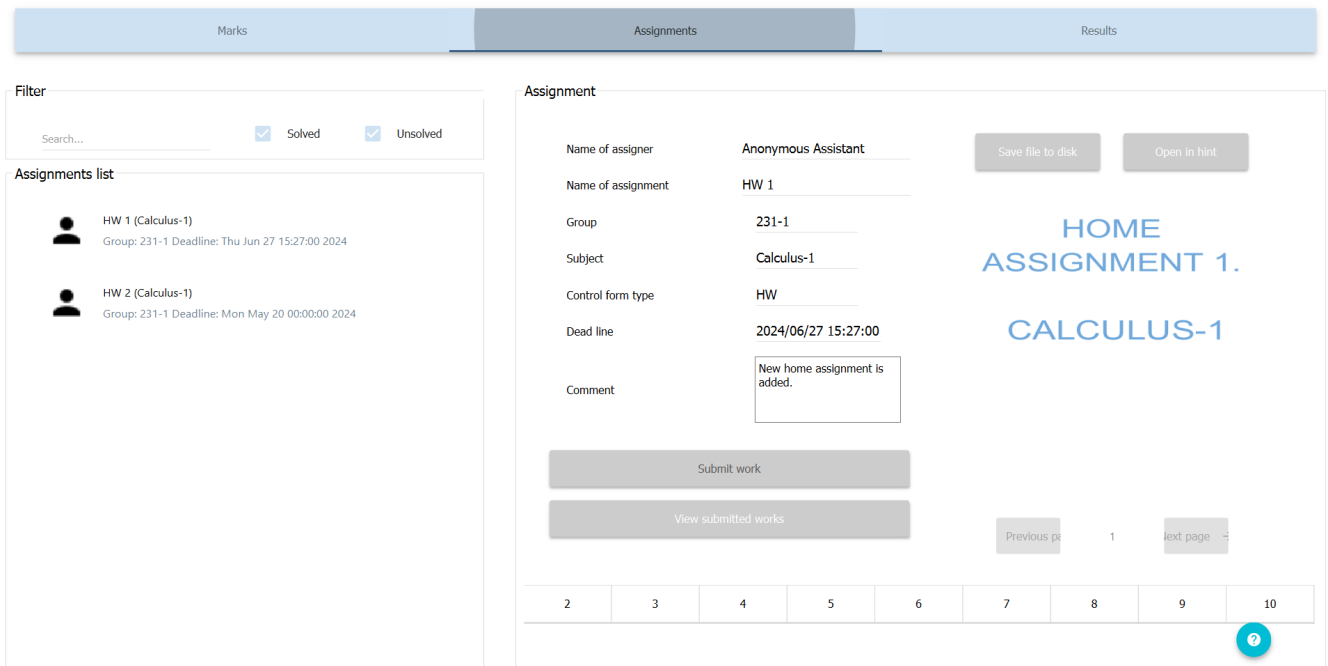


Figure 3.5: Student assignments window.

assignments which can be filtered by existence of submission. When clicking on it the information on the right is changed. There information about the assignment is stored and it is possible to submit a work (3.6):

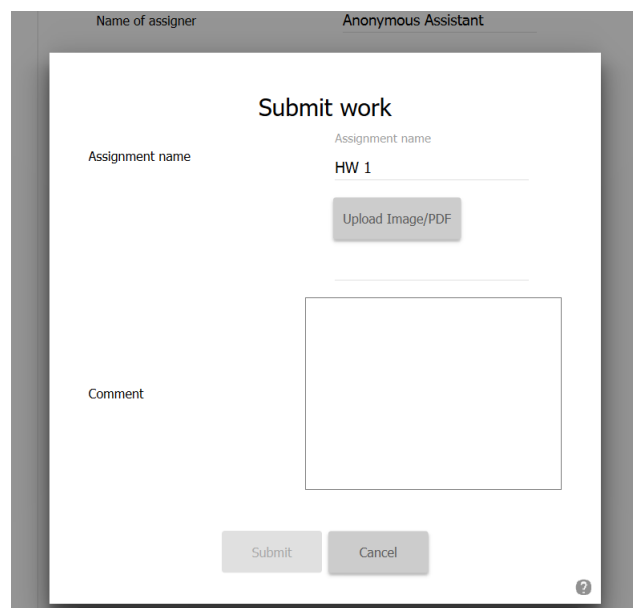


Figure 3.6: Student submission window.

The information about grading can be seen on the bottom table (if it is filled by the assistant) or via looking "Submitted works".

Another thing which can be done here is to ask for consultation (3.7) (on the right bottom corner the blue sign): Finally, the information about final grades is placed in the "Results" window (3.8).

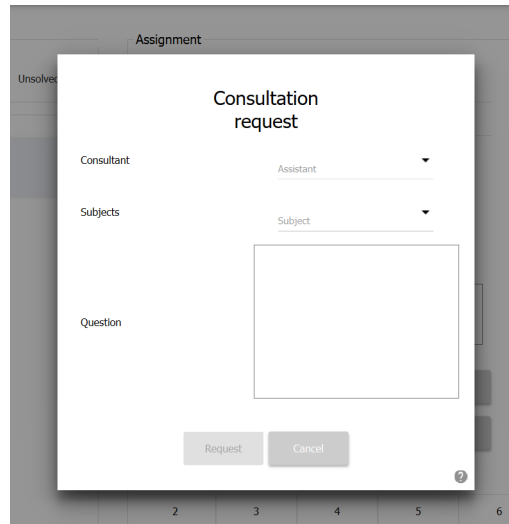


Figure 3.7: Consultation request.

According to the formula specified by the teacher grades for the semesters and final grades are calculated. Also it is possible to enter "Test" values to find out which mark will be received.

Stage	Time period	Element of control	Value	TestValue
Stage 1	1	1st Module	HW	0
	2	1st Semester	HW	0
	3	2nd Module	HW	0
	4	2nd Module	HW	0
	5	1st Module	Quiz	0
	6	2nd Module	Laboratory	0
	7	1st Module	Bonus Task	0
	8	1st Semester	Presentation	0
	9	2nd Module	Laboratory	0
	10	1st Module	HW	0
Stage 2	1	HW	0	
	2	HW	8	
Stage 3	1	1st Semester	Formula	0
	2	2nd Semester	Formula	8

Result: 0 Result: 8 Result: 8

Figure 3.8: Final grades window.

2. Teaching assistant.

The first window (3.9) shows grades of students which are under control of the assistant. On the left there is filter by student. The assignment window looks similar to student's one

The screenshot shows the Assistant window interface. At the top, there is a navigation bar with three tabs: "Marks", "Assignments", and "Consultations". Below the navigation bar, on the left, is a user profile for "Anonymous Assistant" (Assistant, 231-1). To the right of the profile is a search bar labeled "Enter search text" and a "Reload" button. Below the search bar is a list of student filters, including "Example Student" and "Snezhana Klevtsova". The main content area displays a table with the following data:

	Name of a student	Login of a TA	Assignment's name	Subject	Element of control	Mark	Time of adding mark	Date	Comment
1	Snezhana Kle...	assist_1	HW 1	Calculus-1	HW	8	17:41:35	Mon May 27 2...	Very good

Figure 3.9: Assistant window.

but the teaching assistant also has the right to grade submitted works.

Consultation requests send by students can be viewed in the "Consultations" window (3.10):

The screenshot shows the Assistant window interface with the "Consultations" tab selected. The top navigation bar includes "Marks", "Assignments", and "Consultations". Below the navigation bar, there is a "Filter requests" section with a search bar and two checkboxes: "Viewed" (checked) and "Not viewed" (checked). Below the filter section is a "Requests list" containing several entries, each with a student profile icon, name, and subject/assignment details. The first entry is highlighted. To the right of the "Requests list" is a "Consultation request" section showing a student profile for "Example Student" (Student, Data Science and Business Analytics, 231-1) and a request text: "Example, at Wed May 22 16:00:00 2024, ask: Can you please explain HW 1?". A "Mark as viewed" button is located at the bottom right of the request section.

Figure 3.10: Assistant window.

3. **Teacher.** A teacher has the same windows as a teaching assistant; there are only two differences: 1) in the "Marks" window he can see and filter grades by teaching assistants (who are under his control) and 2) if the teacher is a lecturer then he can define a formula (3.11) for his courses: On the left he can choose a subject and for which semester will be this

	Test Value	Variable	Time period	Element of control	
1	1	i1	1st Module	HW	🗑️
2	1	i2	1st Semester	HW	🗑️
3	1	i4	2nd Module	HW	🗑️
4	1	i5	2nd Module	HW	🗑️
5	1	i6	1st Module	Quiz	🗑️
6	1	i7	2nd Module	Laboratory	🗑️
7	1	i3	1st Module	Bonus Task	🗑️
8	1	i100	1st Semester	Presentation	🗑️
9	1	i1000	2nd Module	Laboratory	🗑️

Figure 3.11: Set formula window.

formula. On the right he can define variables (names, time period and element of control). On the bottom he can enter formula and test if it is correct.

4 Technical issues

4.1 Mechanism of the app

Any process in the app consists of several stages. The code is written according to Three-tier architecture ([3]) that allow to track the signal from a user to database.

Consider an example: a user wants to enter the app using his login and password. Firstly, he sees *ui* of main window, where there are *line edits* for user's login and password and *push buttons* "Login" and "Register". After completing fields for login and password, he clicks on the "Login" button sending a request to open the main part of the app for him. The *presentation tier* is finished.

To describe the methods used to implement authentication mechanisms, an abstract interface class *IAuthRepository* has been created, which specifies methods for direct authentication and verification of tokens. Since the implementation of the mechanism on the server and client sides is different, two classes *AuthDbRepository* (used in server (local) or mixed) mode of operation and *AuthNetworkRepository* - used on an application running in client mode (remote) are inherited from the abstraction.

The *AuthNetworkRepository* implementation is actually a proxy class. That is, the application, calling the authentication method using *AuthNetworkRepository*, actually initiates an HTTP REST API request containing JSON with credentials in the body. This request is received by the server part of the application, which, based on the analysis of the HTTP method and URL, determines its identity and passes it on for processing to a chain of implemented REST controllers, each of which processes only those requests that relate specifically to it, otherwise passing the requests further. One of the controllers is *AuthController*, which, having "caught" an authentication request, will transfer it to the *AuthDbRepository*.

The implementation of *AuthDbRepository*, when receiving an authentication request, finds the user's data in the user tables based on the transmitted credentials, determines his role and, if the user is found, generates a special token. The specified token contains information about the time the token was created, the user login of the owner, a specific role, as well as the IP address of the request initiator (will be used in the future to protect against spoofing). This token will be sent in the response about successful authentication and, in the future, will be transmitted by the client for all subsequent requests.

If success, the user will see a screen corresponding to his role. All other processes work in a similar way.

4.2 Security

As it was said earlier, it is essential to secure all data flows to escape cyber fraud and data leakages.

First of all, such important information as password of the user is encrypted by the EnVELOPe (EVP) algorithm. This algorithm is considered to be one of the most reliable. Implementation in C++ is provided by the OpenSSL cryptographic library ([6]). For using the algorithm 256 bit key and 128 bit initialization vector (is needed to prevent repetitions in encrypted texts) were defined.

This algorithm works on the AES (Advanced Encryption Standard) which was developed by the US government. Its main idea is to create a matrix of bytes and rotate it in a specific manner to shuffle the data. Since in our case 256 bit key is used, the matrix will be rotated 14 times. The mode of encryption is CBC (Cipher Block Chaining).

Another issue is using of a token-based authentication ([2]). When a user enters the app and passes his login and password a new token is created and stored (as it was described in the subsection "Mechanism of the app". When a user sends a request next time this token will be found and a method *checkToken* will be called. It checks two things:

- 1) If current IP address differs from the one defined in the token means that either the user changes his location very fast (probably due to using of VPN) or a fraud hacks the user's profile. To stop the second the method will return false and actions in the app will be stopped.
- 2) Checks whether the lifetime of the token is more than a half. If the token has less than a half of the lifetime (in our case less than 30 minutes) and the user is correct then a new token is set (with full lifetime). Otherwise the token is removed from the storage to free the memory.

These methods ensure that a connection is secure and if there will be leakages or fraud attempts the app will recognize them and stop working.

4.3 Technical issues

To make the work of the app better several issues were implemented:

- The app *HSE Grades* is created in the Qt Creator. To store the data the MySQL Workbench was used.

- Multi-thread approach ([4]). Consider the situation: a student clicks on a subject button to view his grades for that subject. The ui of student;s window should display new information simultaneously but because reading of the database is relatively slow the app will freeze for a moment. To escape this creating of parallel threads is needed.

When a request is received usually it is needed to work with the database. To do this methods of *workers* are called. Their mechanism is the following: firstly, a new thread is created using QThread and operations are moved to it. When the thread starts the *process* method starts its work (communication with the database' repositories). When the process is finished and results are sent, the thread is quitted and the worker will be deleted when returning to the main thread (as well as the created thread).

This allows to minimize delays.

- Creating of *DTO models*. DTO (Data Transfer Object) carries the data between processes. In this app all classes of DTO models are used to receive/send information to the HTTP server (so they all can read and write data in JSON type). This type of classes also supports the Three Tier architectural model (the application tier).
- Repository. Classes of this type communicate with the database and send queries to it. There requests for updating, inserting, deleting and receiving data are created. All the data which is received is stored in *models* (like student, teacher, grade etc.).
- Visualization. To make the app more attractive the Qt Material Widgets library was used. It allows to make beautiful effects, animation of buttons and appealing interface. Also a special feature as installing an avatar is supported by the ImageCropper and ImageEdit classes (which use QPainter and QPainter libraries).

5 Conclusion

To conclude with, the *HSE Grades* app provides both students and the teaching staff with convenient interface and all functions needed. The main advantages are:

- The app does not refer to any non-HSE services following the idea of vertical integration of the university;
- Data transmission processes are secure. Encryption and token-based authentication minimize chances of frauds;
- Processes of adding, viewing, submission and grading of assignments is located in one app;
- Students can see both final and intermediate grades;
- Teaching assistants and teachers can create grading tables in the app and spend less time on it than it was before;
- A student can see only his grades in a grading table. This is good for educational purposes (tracking of other students performance can be harmful from the psychological side).

From the other side, there are some features which are not provided in this version of app:

- Opportunity to change language of the app;
- A teacher should have an opportunity to ask a manager via the app to create new assistants profile (so that he can appoint it as a teaching assistant of one of his subjects);
- Opportunity to export grades as a table of .csv format;
- A student can see his distribution of grades through a histogram;
- Make a window for schedule representation;
- Provide access to the HSE online library;
- Make a news block.

References

- [1] CLOUDFLARE. *Main issues about HTTP*. URL: <https://www.cloudflare.com/learning/ddos/glossary/hypertext-transfer-protocol-http/>.
- [2] CLOUDFLARE. *Token-based authentication*. URL: <https://www.cloudflare.com/learning/access-management/token-based-authentication/>.
- [3] GeeksForGeeks. *Three Tier architectural model*. URL: <https://www.geeksforgeeks.org/three-tier-client-server-architecture-in-distributed-system/>.
- [4] JavaTPoint. *Reasons for multi-threading*. URL: <https://www.javatpoint.com/benefits-of-multithreading#:~:text=Multithreading%20allows%20many%20parts%20of,increases%20CPU%20utilization%20through%20multitasking..>
- [5] MySQL. *MySQL official website*. URL: <https://www.mysql.com/>.
- [6] OpenSSL. *Envelope algorithm*. URL: https://wiki.openssl.org/index.php/EVP_Symmetric_Encryption_and_Decryption.
- [7] Qt. *Qt Creator documentation*. URL: <https://doc.qt.io/>.