Report about app

Author: Arturs Rinkis

Institute: TSI

Course: 3

Group: 4402BD

Language: Russian, but this project made in English

Riga, 2017

1. Introduction

2. Description develop applications

3. Application architecture

4. The choice of tools and development tools

5. Database development

5.1. The definition of the entities

5.2. Conceptual design

5.3. Logical design

5.4. Physical design

6. The development of server-side

6.1. The data access layer

6.2. Level business services

6.3. A web service

7. Development of client-side

7.1. The interaction with the server

7.2. Implementation of Android application

8. User manual

9. System requirements

10. Bibliography

**1. Introduction**

The relevance of the research topic. Knowing their ranking among classmates on any subject in the school is essential to the success of the student learning experience.

At the same time, the use of the application "University Student Rating", allows you to save time of the teacher for entering the message and needs assessment to the student.

Thus, the relevance of the research topic due to the computerization of the student and the automation of action for the placing of a teacher a teacher.

The aim of this work is to develop an Android application capable of providing ease of use and the level informatization student and automate actions in placing the teacher as a teacher.

In accordance with a purpose, in front of the following tasks:

*1. to Consider the basic principles of dimensioning evaluation of teachers and principal ratings for students on the obtained evaluations.*

*2. To perform possible solutions to this problem and choose the most optimal method of implementation tasks for the Android OS.*

*3. Database design, based on the obtained data.*

*4. Design and development of backend applications.*

*5. Designing and developing client part of the application.*

**2. Description develop applications**

The development application is for students and teaching staff of Universities. Main purpose of the application – evaluation of students by teachers, viewing their rankings of students and compare their own rating with the ratings of other students.

The application involves the separation of ratings on items and separation of items according to types of human activity:

*1. Science – mathematics, physics, Informatics, biology, chemistry and other.*

*2. Art – visual arts, music, literature and other.*

*3. Sport – physical culture and sport.*

*4. Society – psychology, history, social studies.*

The rating also divided into types:

1. Classwork – evaluation for laboratory and course works. The assessment will be billed in accordance with the points system in the school (a 10-point system).

2. The Olympics and the scientific and other achievements the assessment of achievements in extra-curricular activities, competitions and other achievements the assessment of which does not fall into the framework of score-rating system.

3. Penalty points – points obtained by a student for any infraction: absenteeism, behavior, or untimely execution of the tasks of the teacher and so on.

Rating calculations are also divided into types. Rating average score on classroom work, according to the total score for the Olympics and for the overall assessment, which consists of all types of ratings, with the deduction of penalty points.

The functionality of the application is different for students and teachers:

1. The functionality for students.

• The profile tab. Contains basic information about the student: General statistics, the average score for all subjects and personal data. The profile of the student will need to indicate teachers who have the right to judge it.

• Tab rating. Contains information about the rating of the student for the selected subject. The rating is calculated as for the individual student, and in comparison with other students of the University and the course.

• Tab competitors. Is a search for students who compete on the rating for this student. There is a filter.

• The search tab. Is a conventional user search filter.

2. Functionality for teachers.

• The profile tab. Contains basic information about teachers: the average score of all his students and personal data. The profile of the teacher must specify the subjects which he teaches.

• Tab assessment. Allows the teacher to evaluate the student, and view and delete the previous assessment.

• The search tab. Is a conventional user search filter.

**3. Application architecture**

The application must provide students and teachers the opportunity of registration with the indication of their data as well as authentication and authorization from any device with a username and password. These requirements is possible only in the case of a client-server application.

To store and retrieve data, and modify data authorization, which involves the use of the server. These requirements are possible with the use of DBMS.

Because system components have to handle identical data and share them, it is necessary to use a method of data transfer between client and server. In this way, a pattern Data Transfer Object (DTO). Use DTO suggests a lack of availability implement behavior that allows it to be transmitted between subsystems in the JSON format.

In the end, you must implement the 3 subsystems of the application:

1. DBMS for storage of user data.

2. The server part is for database connection, authentication and authorization of users, and implement the logic of data processing and transmission to the client of the result (the calculation of the rating, search, etc.).

3. The client part – implementation of the graphical user interface, receive and output data from the server.

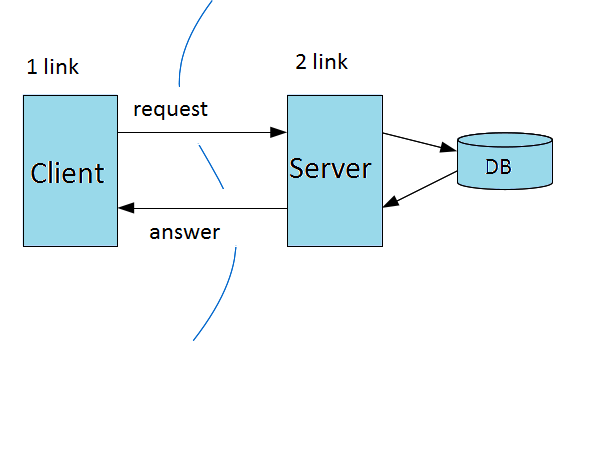
Schematically, the application architecture can be represented as follows:

Figure 1. Schematic representation of the application architecture. (

**4. The choice of tools and development tools**

To design DBMS and server side is to define the platform for which you want to develop these components.

Such a platform is Windows, which supports a large number of DBMS and is a server platform with the necessary tools to develop under IIS: WCF ASP.NET.

For Windows standard DBMS is MSSQL, which was chosen for the development of DBMS.

To develop backend was chosen as the development environment is Visual Studio 2013.

Unlike the backend, the platform for the client side is known in advance of Android. The preferred development environment, which is the Android Studio.

In the End, based on the selection of platforms, databases and development environments, we can conclude that for writing the backend will be used WITH YAP#, and the development will be conducted based on the framework ASP.NET MVC5 with the use of technology to create web services – WebAPI2. For the client part is used JAPANESE Java using standard development tools provided by Android Studio Android.

**5. Database development**

**5.1. The definition of the entities**

Entity – any concrete or abstract object in the subject area. Entities are the basic types of information that are stored in the database (in a relational database each entity is assigned to the table).

In the application, "University Student ratings" in the design were allocated to the following entities:

• City

• Country

• Course

• Discipline

• DisciplineBranch

• Faculty

• GenderType

• ProfessorDiscipline

• Rating

• RatingDivision

• RatingSystem

• RatingType

• Status

• StudentProfessorDiscipline

• University

• UserProfile

**5.2. Conceptual design**

The main feature of a DBMS is the availability of procedures for entering and storing not only the data but also descriptions of their structure. The files contain a description of the stored data under a DBMS, became known as data banks, and then "Database" (DB).

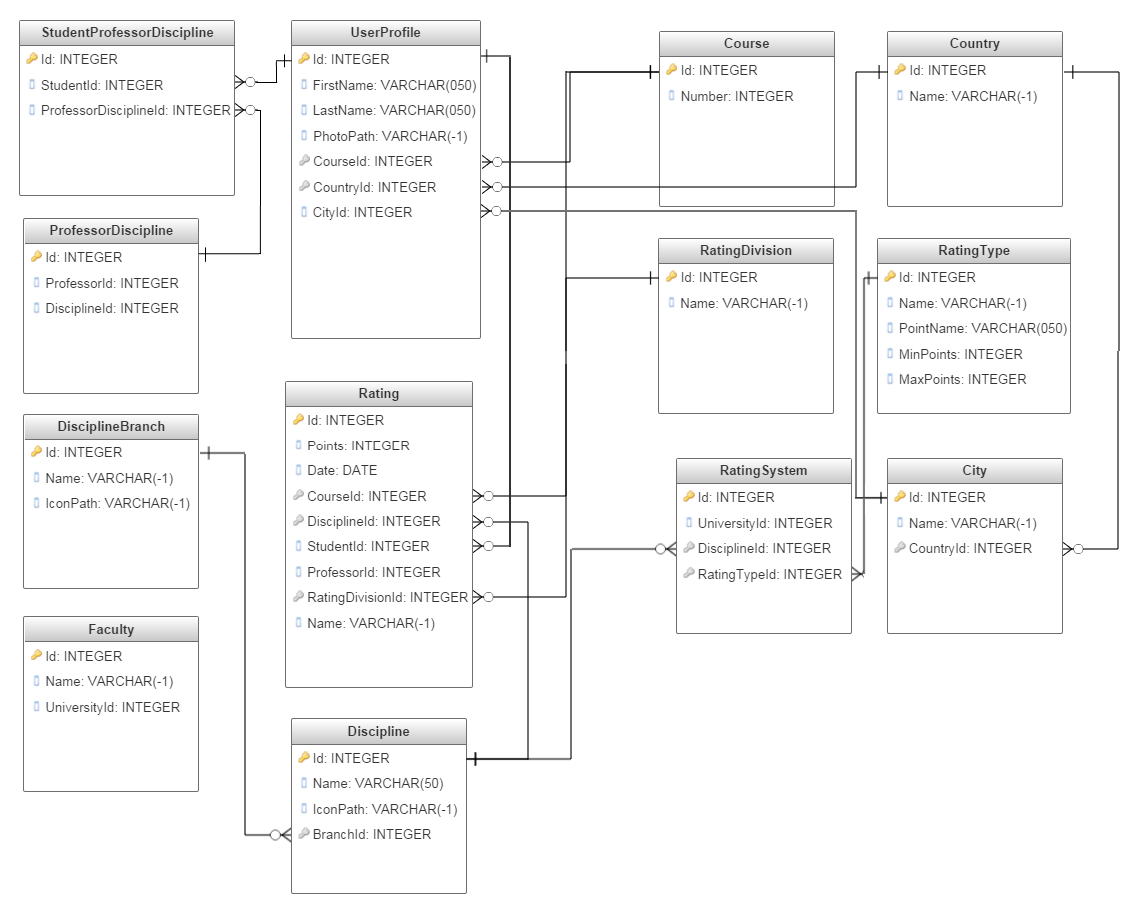
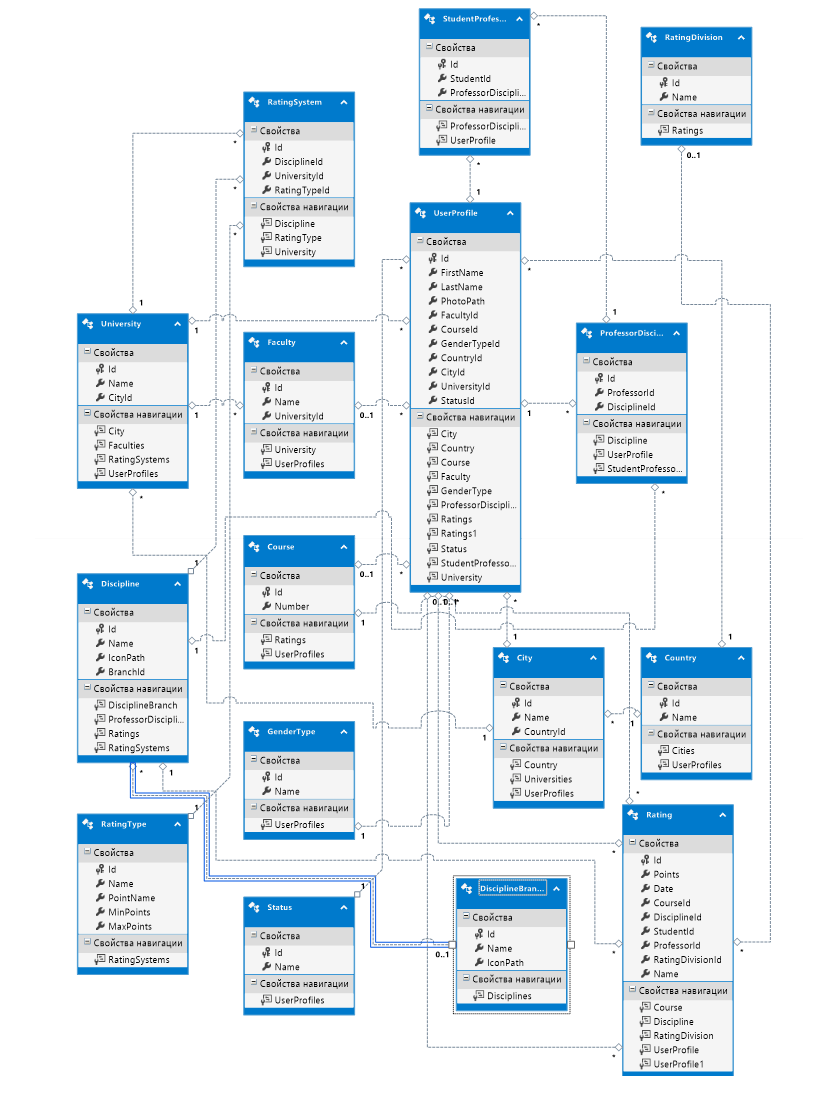
Conceptual model-is tematicheskaya domain model on a conceptual level, not tied to any DBMS, and a computer.****

Figure 2. Diagram database at the stage of conceptual design

**5.3. Logical design**

**Logic model -** a diagram of the operation of the database based on specific data models, e.g. relational data model. For the relational data model datalohic model — a set of relationship patterns, usually with indication of primary keys and also the "ties" between the relationships representing foreign keys. The diagram on this step corresponds to the diagram at the stage of conceptual design

**5.4. Physical design**

Physical data model describes the logical implementation of the object model object-level database-specific. Figure 3. Diagram database at the stage of physical design

**6. The development of server-side**

As a client-server app, the server part is also divided into various links. The separation is based on the popular for such applications, N-tier architecture (in this case three-tier). The system is composed, respectively, of the 3 levels:

• DataAccessLayer (data access layer). The level contains classes for working with the database, classes that implement the authorization system, the classes of data access from higher levels.

• ServiceLogicLayer (business logic layer). Level processes the data retrieved from the database, creates if required the DTO objects, and provides the web service data transmitted to the client application.

• WebServiceLayer (web service). The tier handles requests from the client, authenticates and authorizes the user, transfers the customer data from the services.

* 1. **DataAccessLayer**

The level includes the following classes and interfaces:

1. USR\_DBEntities – context access to data. The main class to work with databases based on the ORM EntityFramework, making basic work with the database is happening inside ORM that allows to abstract from a DB. The class provides access to database tables is described in the previous section. The context also allows to automate the work with the authorization system ASP.NET Identity through inheritance from the class IdentityDbContext

2. ApplicationUser, ApplicationRole, ApplicationUserManager, ApplicationRoleManager classes to initialize and operate the authentication system ASP.NET Identity, explanation of the working principle and mechanism of action of which is beyond the scope of this report.

3. IUnitOfWork interface to implement the unit of work pattern. UnitOfWork serves one purpose: to have confidence that when using multiple repositories that work with the database, work was carried out with the same DbContext instance.

4. UnitOfWork implementation of the IUnitOfWork interface.

5. IGenericRepository interface to implement generic repository pattern. The idea of the pattern repository is to create a level of abstraci between data access layer and the business layer. This will help to isolate the application from changes in the data warehouse and will allow you to create tests for some units.

6. GenericRepository implementation of the IGenericRepository interface.

* 1. **ServiceLogicLayer**

The level includes the following classes and interfaces:

1. AutoMapperConfig class for configuration library AutoMapper

2. Classes DTO classes to transfer data of the client part or receiving data from the client. Objects of classes serialized/deserialized from/to JSON format.

3. IAuthService Interface for a service that describes methods for working with the authorization system.

4. AuthService – interface Implementation IAuthService that implements these methods using the authorization system ASP.NET Identity.

5. IDataService Interface for a service that provides methods to work with data from the database.

6. DataService implementation of IDataService service.

7. ISearchService Interface for a service that describes methods for searching.

8. SearchService – the service implementation ISearchService.

9. IStatisticService Interface for a service that describes methods for information about ratings.

10. StatisticService – service implementation IStatisticService.

11. IUserService Interface for a service that describes methods for obtaining information about users.

12. UserService implementation of IUserService service.

13. IServiceCreator interface for the implementation of a pattern "Abstract factory" for creating services.

14. ServiceCreator the implementation of the abstract factory based on the described classes for services.

* 1. **WebServiceLayer**

The level includes the following classes and interfaces:

1. AccountController – WebAPI2 controller. Contains API methods to register, authenticate, exit and check whether the user is authenticated.

2. DataController – WebAPI2 controller. Contains API methods that provide public information about faculties, courses, countries, cities and other methods. Also allows you to upload images from the server.

3. HomeController – ASP.NET MVC5 controller. Controller to list the API methods from a web browser. (aid for testing and development).

4. SearchController – WebAPI2 controller. Contains the API methods to search for users and competitors.

5. StatisticController – WebAPI2 controller. Contains API methods for retrieving data about the ratings of students.

6. UserController – WebAPI2 controller. The API contains methods to retrieve and modify data about users.

**7. Development of client-side**

The client part of the application, as well as the server part consists of different parts, which can be subdivided into classes to interact with backend applications through the mechanism of request-response HTTP Protocol and classes to interact with the Android OS and implementing a graphical user interface for this OS.

**6.2 The interaction with the server**

The level includes the following classes and interfaces:

1. Classes DTO classes to transfer data of the client part or receiving data from the client. Objects of classes serialized/deserialized from/to JSON format.

2. IAuthService Interface for a service that describes methods for working with the authorization system.

3. AuthService – interface Implementation IAuthService that implements these methods using the authorization system ASP.NET Identity.

4. IDataService Interface for a service that provides methods to work with data from the database.

5. DataService implementation of IDataService service.

6. ISearchService Interface for a service that describes methods for searching.

7. SearchService – the service implementation ISearchService.

8. IStatisticService Interface for a service that describes methods for information about ratings.

9. StatisticService – service implementation IStatisticService.

10. IUserService Interface for a service that describes methods for obtaining information about users.

11. UserService implementation of IUserService service.

12. IServerAPIManager interface for the implementation of a pattern "Abstract factory" for creating services.

13. ServerAPIManager – implementation of abstract factory on the basis of the described classes for services.

14. CookieManager class for working with cookies. Because server saves the state of the user by storing a cookie set on the client storing and receiving cookies is implemented using Preferences, for which he meets the class.

15. HttpRequestUtility class that implements methods for sending requests to the server.

**6.3 Implementation of Android application**

To implement Android application was created 6 activities and 13 fragments. For correct display your data has established its own controls (View) and adapters.

We implemented the following basic classes of activities, fragments, and custom views and adapters for them:

1. WelcomeActivity Activity displayed when the application starts, if the user is not authenticated. If the user is authenticated (this procedure )

2. LoginActivity Activity to login using the login/password.

3. SignUpActivity Activity for new user registration.

4. GuestActivity Activity to view the application by an unauthorized user.

5. StudentActivity activity for the authenticated user with the role "student"

6. ProfessorActivity activity for the authenticated user with the role "Professor"

7. ProfileFragment – fragment to display basic information about the user with the role "student"

8. ProfessorProfileFragment – fragment to display basic information about the user with the role "Professor"

9. ProfessorDisciplinesFragment – fragment to display the list of subjects taught by the teacher disciplines

10. ProfessorRatingFragment – snippet to display the data indicated by the teacher assessments for the selected discipline of the previous fragment. It is also possible to delete grades, and placing the new assessment.

11. ProfessorSetRatingFragment – fragment for placing the evaluation by the teacher.

12. RatingDisciplinesFragment – fragment to display the list of courses a student

13. RatingDisciplineFragment – fragment to display the data on the rating for the selected discipline of the previous fragment. The rankings are presented in Tables, graphs and textual information.

14. ConcurentFragment – fragment to display the list of competitors for the currently logged student.

15. ConcurentSettingFragment – fragment to set a filter which will display the previous competitors for the described fragment.

16. SearchFragment fragment to search for users according to the filter.

17. SettingSearchFragmen t– piece to the filter settings which will be displayed for users of the previous described fragments.

18. ShowProfessorProfileFragment a fragment to view the teacher profile, not a profile of the current authorized teacher.

19. ShowStudentProfileFragment a fragment to view the student's profile, not a profile of the current authorized student.

20. NonScrollListView – your own control to create a non-scrollable list view. This control is necessary to display the select teachers (for student) or subjects taught (for teachers) in the fragment of the user profile.

21. SquareImageButton – your own control to create a square ImageButton. This control is required to implement buttons: Login, SignUp, Guest, Close the app in the WelcomeActivity.

22. SquareImageView – the same goal as the previous View, only ImageView. Used in various parts of the application.

23. BranchAdapter, ConcurentResultAdapter, DisciplineAdapter, ProfessorDisciplineAdapter, ProfileDisciplineAdapter, SearchResultAdapter, StudentRatingAdapter adapters for the corresponding ListView.

1. **User manual**

Main screen description

Immediately after launching the application displays the main application screen with a choice of further actions:

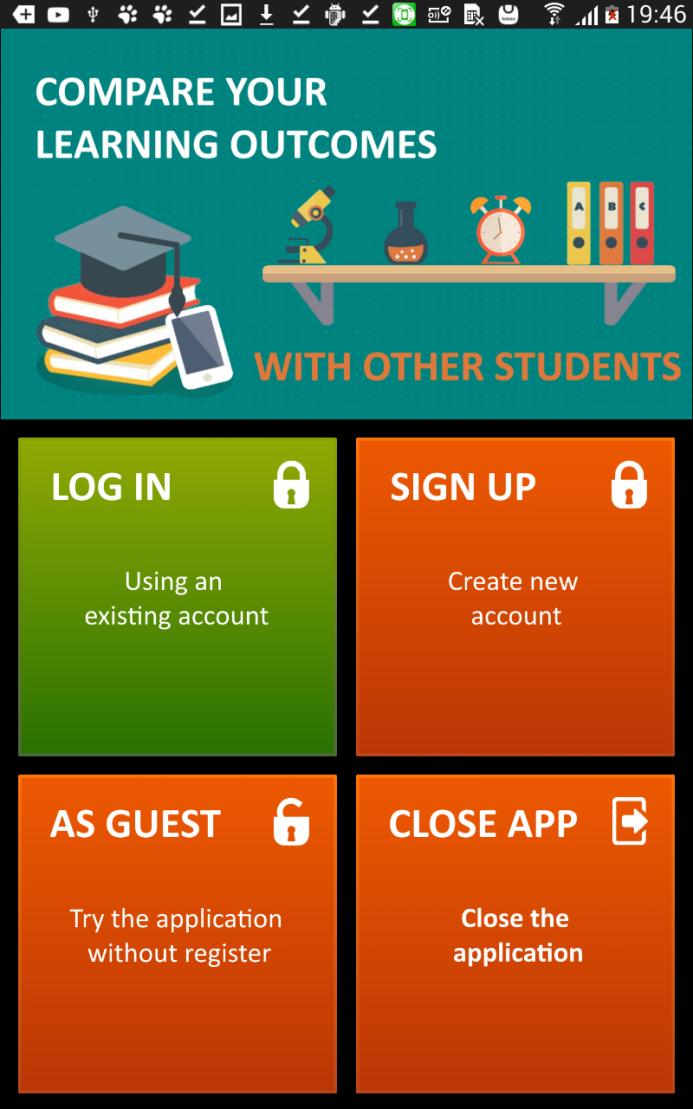
Login password registration, login guest, exit the application.

Figure 4. The main screen of the app.

1. Registration

Register a new user.

To register a new user, you must enter the required data, then click the "REGISTER"button.

If data is entered correctly, you will see a profile screen. Otherwise, you will receive an error message at the top of the screen to register.

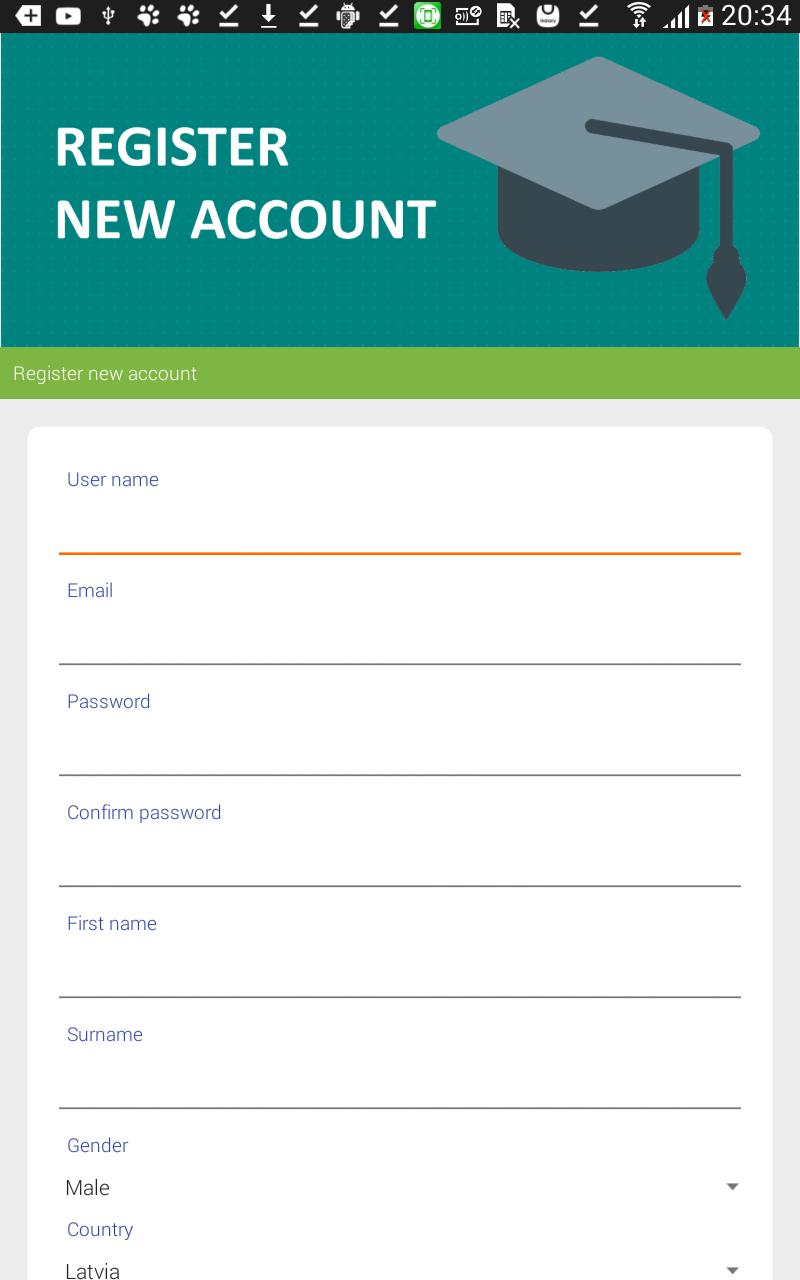


Figure 5. Registration.

1. Entrance.

To login the user must enter a username/password of a registered user.

After login screen will be displayed user profile.

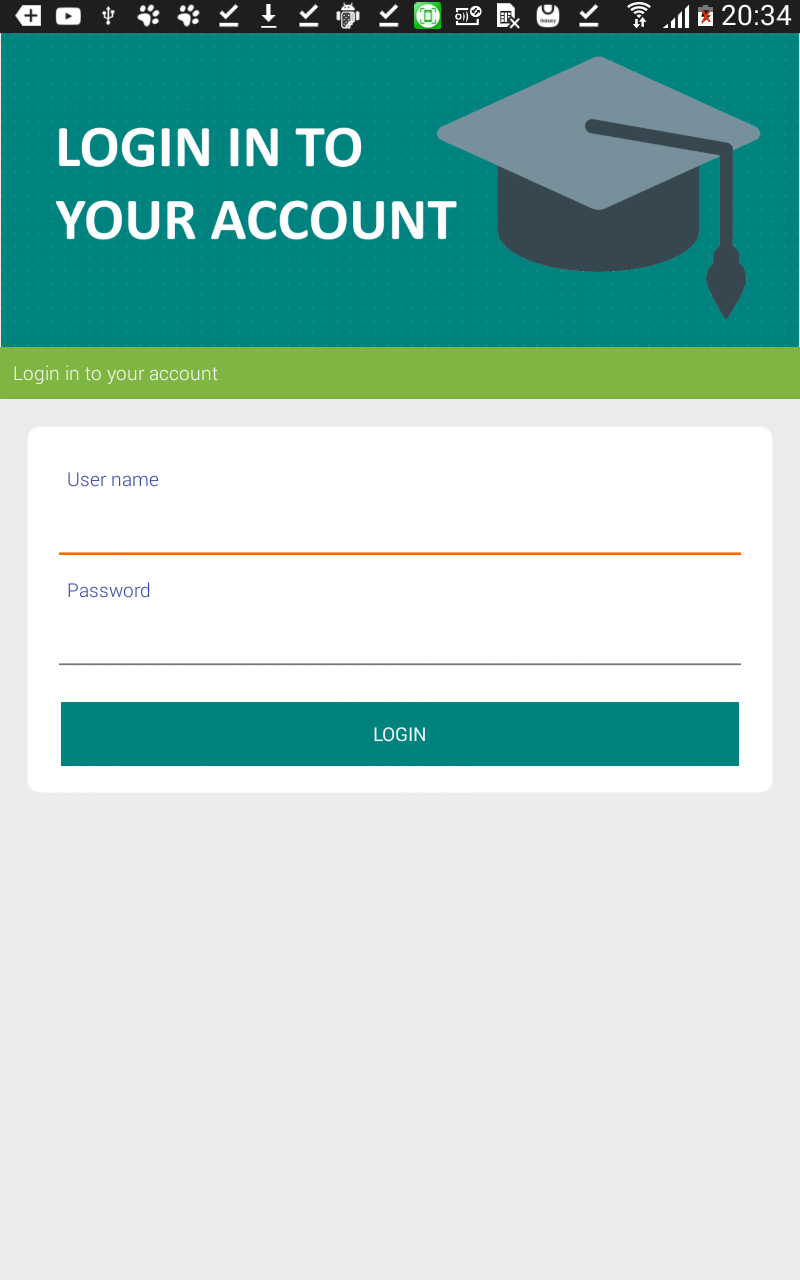


Figure 6. Entrance.

1. Profile description for the student.

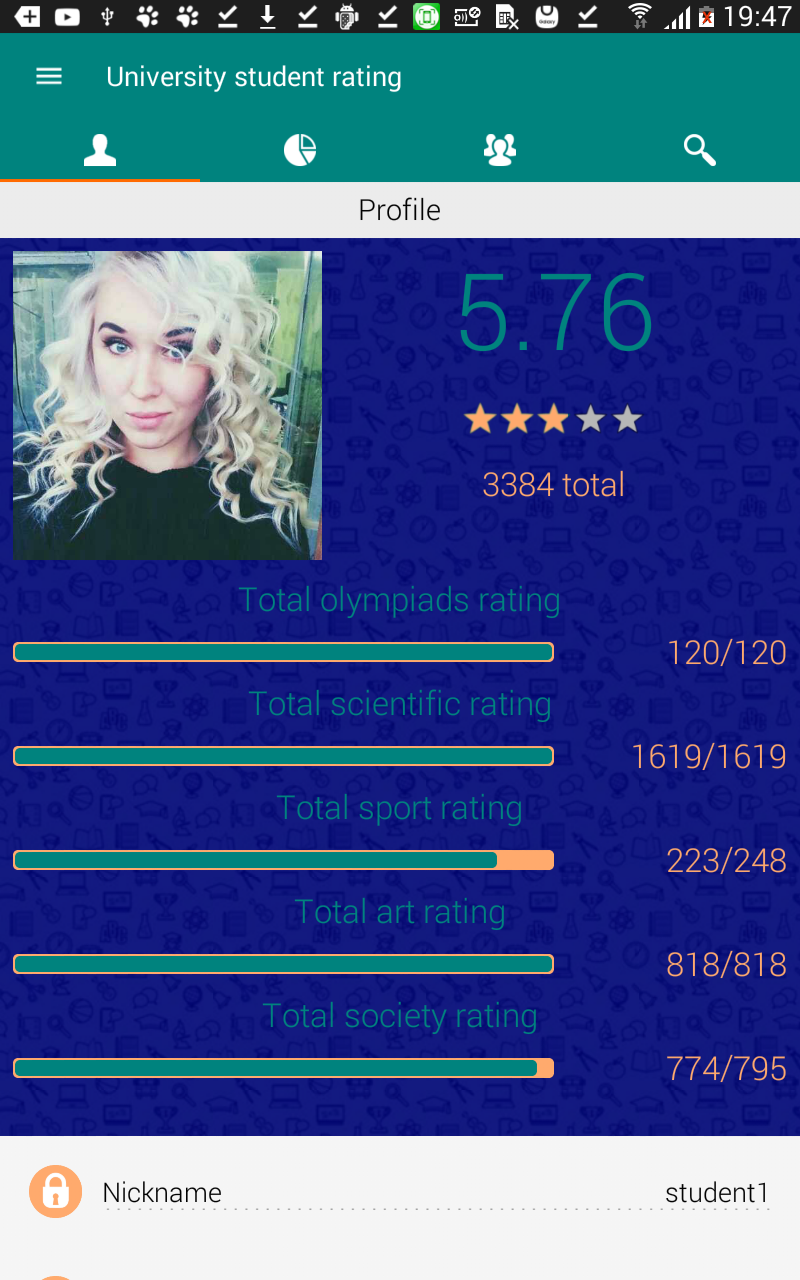
In the student's profile you can see the basic data about the student and the list of selected teachers who have the right to put it with.

Figure 7. The profile of the student.

1. Description statistics.

To view statistics the user must click on the 2nd tab on the screen. Next, you need to select the direction of the object and then select the object itself.

Statistics presented in various tables and graphs, the data are intuitive.

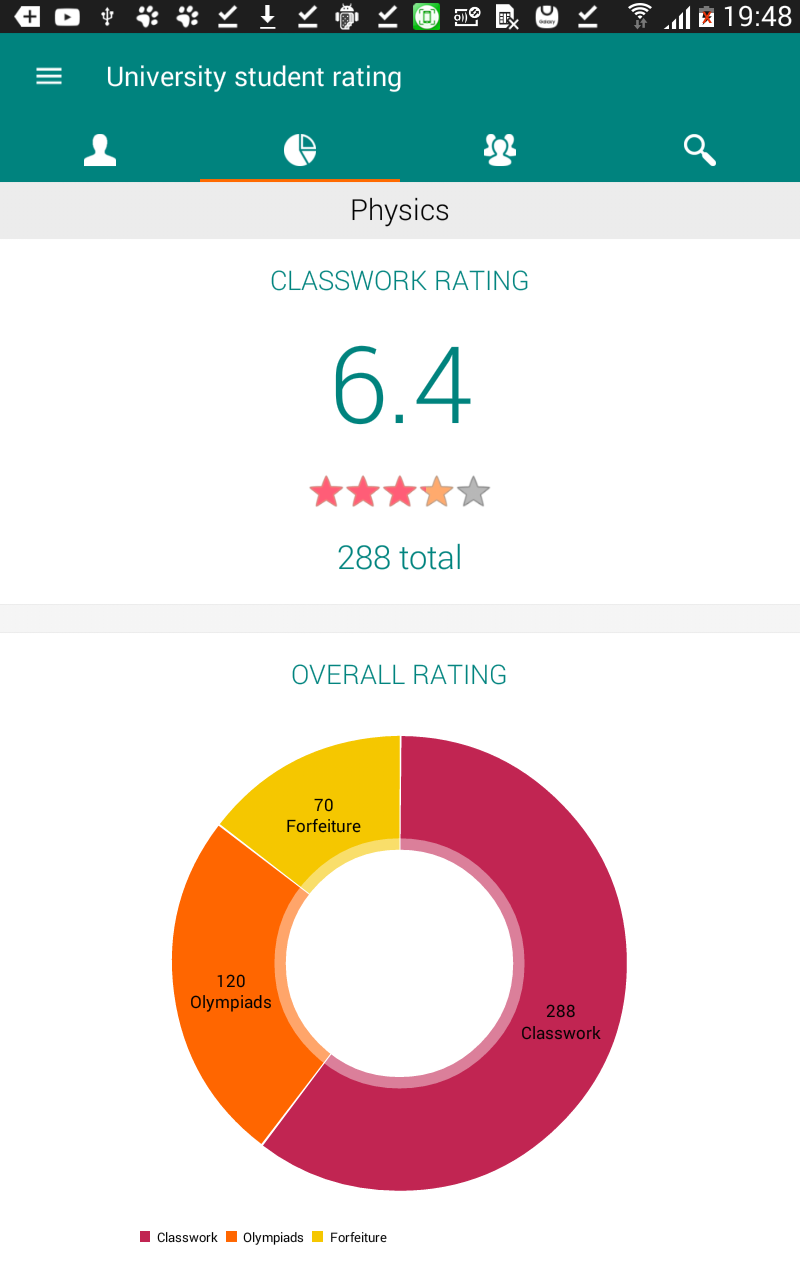


Figure 8. Statistics student.

1. Description of competitors.

To view the list of competitors you need to click on 3rd tab on the screen. You will see a filter which is searched competitors. After clicking on the "SEARCH" button you will see the list of competitors.

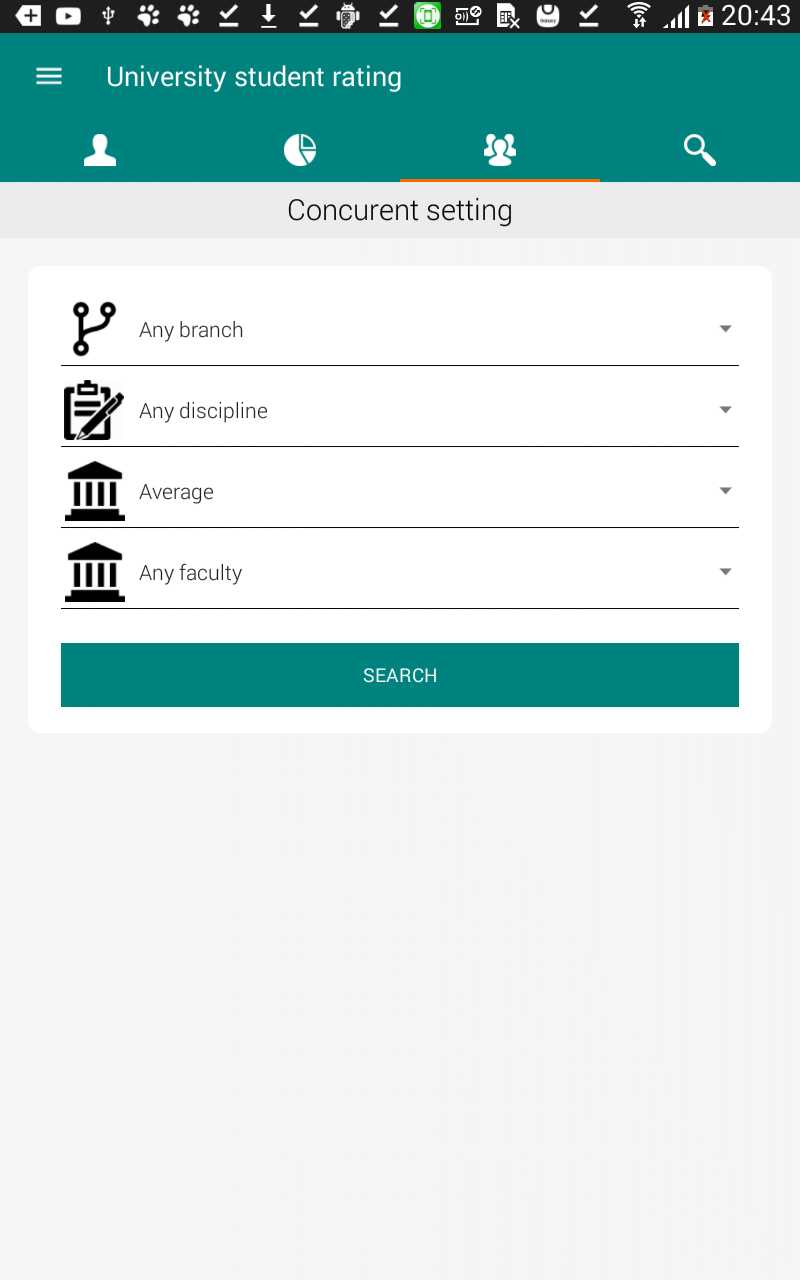


Figure 9. Search competitors student.

1. The description of the search.

Search identical to the list of competitors, with the exception that the view of the competitors is relative to the current authenticated user, and search on all users.

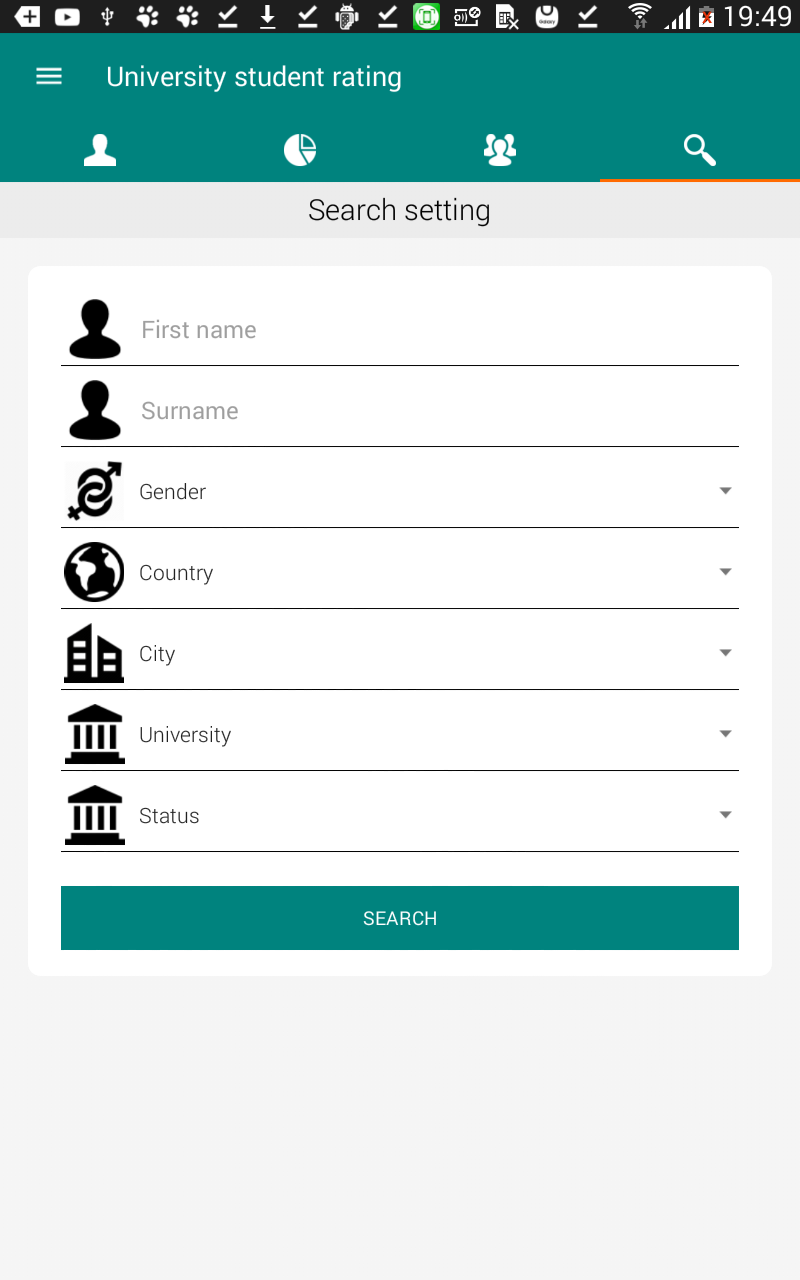


Figure 9. Search users.

1. The description of the input by the guest.

Entrance visitor allows only users with viewing their profiles.

2. Description of the profile piece for the teacher.

The teacher profile you can see the basic data about the teacher and a list of chosen subjects which he teaches.

3. Description of view/placing the rating of the teacher.

For viewing and placing the rating of the students need to click the 2nd tab. Opportunities: view marked assessments, delete assessments, setting assessments to the student. For input evaluation, you must choose appropriate a student who is enrolled at an authorised teacher.



Figure 10. Viewing/entering the rating.

1. **System requirements**

To deploy the server application ASP.NET MVC5 need hosting with support for IIS7.0 and above, also support MSSQL version 2012 and above.

To run the client application, you need a device that supports the Android operating system version 4.4.2 (KitKat) and above, as well as access to the Internet and free space is ~10MB.