Politechnika Wrocławska

Faculty of Computer Science and Management
Field of study: **COMPUTER SCIENCE**

Bachelor Thesis

Web-based educational game with gamification supporting the work of the teacher, with RWD and flat design, on selected technology platform.

Aleksandra Magdalena Janik

keywords:
web application
gamification
educational game

short summary:
The purpose of this work is to design and implement application for students in classes from 4th to 6th in primary school that will help them in math-learning process. Within the work gamification mechanisms will be studied, overview of existing solutions will be done and designing and implementing process will be described.

Supervisor	doc. dr inż. Jacek Gruber	.....	…….
	Title/ degree/ name and surname	*grade*	*signature*
The final evaluation of the thesis			
Przewodniczący Komisji egzaminu dyplomowego	.....	.....	…….
	Title/ degree/ name and surname	*grade*	*signature*

For the purposes of archival thesis qualified to: *

a) Category A (perpetual files)
b) Category BE 50 (subject to expertise after 50 years)

* Delete as appropriate

stamp of the faculty

Wrocław 2017

Abstract

The purpose of this thesis is to design and implement web-based educational game that will allow teachers to use new method for encouraging pupils to learn mathematics. Idea for the topic came from the visible need to help children in learning process. Gamification mechanisms were studied as well as overview of existing solutions was done. Functional and non-functional requirements were formulated and transformed into use case specifications. Frontend of the application was created mainly with the use of HTML and CSS, and using Bootstrap allowed to implement system that is responsive. Backend of the application was created using PHP. Methods used to test the application were described and user's instruction manual was provided.

The result of this work is an application that can be used by students in a class as new and interesting study help.

Streszczenie

Celem tej pracy jest zaprojektowanie i implementacja internetowej gry edukacyjnej, która umożliwi nauczycielom wykorzystanie nowej metody zachęcania uczniów do nauki matematyki. Pomysł na temat pracy powstał z widocznej potrzeby pomocy dzieciom w procesie uczenia się. Zostały przeanalizowane mechanizmy grywalizacji oraz wykonano przegląd dostępnych rozwiązań. Wymagania funkcjonalne i niefunkcjonalne zostały sformułowane i przekształcone w scenariusze przypadków użycia. Frontend aplikacji został stworzony głównie z wykorzystaniem HTML i CSS, a użycie Bootstrap pozwoliło na zaimplementowanie systemu, który jest responsywny. Backend aplikacji został utworzony przy użyciu PHP. Opisano metody użyte do przetestowania aplikacji oraz dostarczono instrukcję użytkowania.

Rezultatem tej pracy jest aplikacja, która może być wykorzystana w klasie przez uczniów jako nowa i ciekawa pomoc naukowa.

Table of Content

1. Introduction 7
   1.1. Purpose and scope of work 7
2. Vision 8
   2.1. Description of application 8
   2.2. Positioning 8
      2.2.1. Problem statement 8
      2.2.2. Product position statement 9
   2.3. User summary 9
   2.4. Gamification 9
      2.4.1. Theory 9
      2.4.2. Usage in application 12
   2.5. Existing solutions 13
      2.5.1. Learning Apps 13
      2.5.2. Khan Academy 14
      2.5.3. Kahoot! 16
      2.5.4. Duolingo 17
      2.5.5. Squla 19
      2.5.6. Comparison between 20
3. Application design 21
   3.1. Requirement specification 21
      3.1.1. Functional requirements 21
      3.1.2. Non-functional requirements 22
   3.2. User interface 22
      3.2.1. Responsive web design 22
      3.2.2. Flat design 23
   3.3. Use-case diagram 24
   3.4. Use-case specification 25
   3.5. Database schema 36
4. Implementation 38
   4.1. Relevant technologies 38
      4.1.1. Frontend 38
      4.1.2. Backend 42
      4.1.3. Database 42
   4.2. Mechanism of questions and answers displaying 42
      4.2.1. Training mode 42

4.2.2. Competition mode 43
4.3. Quality assurance 45
4.3.1. Manual tests 45
4.3.2. Tests automation 45
5. Instruction manual of the application 47
5.1. Instruction manual for Student 47
5.1.1. Training mode 49
5.1.2. Competition mode 51
5.1.3. Statistics 53
5.2. Instruction manual for Teacher 54
5.2.1. Manage questions 54
5.2.2. Statistics 56
5.3. Instruction manual for System Administrator 56
5.3.1. Manage users 57
6. Summary 59
6.1. Conclusions 59
6.2. Future work 59
Bibliography 60
List of figures 61
List of tables 62
Attachments 63

1. Introduction

1.1. Purpose and scope of work

My mother is a math teacher in a primary school. Thanks to that I keep hearing about how kids nowadays have lack of motivation to learn. They don't feel the need to receive a good grade nor do they understand that knowledge which teachers want to transfer will be useful for them in their future life. Mastering those basic topics that are thought in primary school is also necessary on further stages in their education. Without that they will always have to catch up on the material instead of taking full advantage of the lessons.

The most important problem is that the topics, especially in math, but difficult for many pupils, but they don't want to practice because they think it is boring. For them spending their time after school on solving equations is not what they would choose, but in many cases time that is destined for math lessons at school is not enough. Instead of doing homework they prefer to go outside and spend time on the free air playing with friends or, more and more often, spend this time on the computer.

Now kids are staring using computers earlier than before. According to Gemius Polska, in Poland 9.4 percentage of people who are using Internet is between age 7 and 12 [7]. Moreover, the same research showed that they most often visit webpages with online games and those designed especially for young audience.

There is a way to combine what children want to do with what they should do – educational games. Joe Peters in his article presents following advantages of using computer games in teaching process: [15]

- Motivation and engagement
- Immediate feedback
- Cognitive growth
- Digital literacy
- Skill development

However, how to keep users to play the game and not become quickly bored with it? Marc Robinson from GameAnalytics said that "*On average, less than 40% of players return to a free-to-play game after just one session*." [13]. That is way when designing a computer game, first point from above list, motivation and engagement, needs to be fully met. The solution here is gamification.

I designed my application with the thought of bringing the teachers additional study aid which will help them motivate and interest pupils with the topic and will allow to check kids' knowledge in easy way. I decided to focus on mathematic teachers and children from classes from 4 to 6 in primary school, as mathematic is one of the most trouble-causing subject and at the same time one of the most important ones.

Scope of the work includes designing and creating a responsive educational application with elements of gamification together with the extension of gamification topic and overview of existing solutions.

2. Vision

2.1. Description of application

Wanting to create an application that will serve both students and teachers, I decided to do it in a form of a quiz. This way teachers in easy way will be able to add more questions and expand the game as well as adapt it to their needs.

In my application I want user to have two modes to choose from:

- training
- competition

When selecting training mode, after choosing category, player will be able to select number of questions on which he wants to test his knowledge. When he will select correct answer progress bar will show that, and if wrong one, the question will be asked again so that at the end of the training user answers all the questions correctly.

Competition mode is destined for two players. First, login of the second user is needed. Each player will be asked five questions with the limit of time for all the answers set to five minutes. After the competition players can see how many points they received, where every correct answer gives 10 points and additional 50 points for player who wins the competition. At the next page users can also see competition questions with correct answers and answers selected by them. Points are added for both players and if specified amount of points is reached, user goes to the next level.

In addition, users can also see statistics to check how many points they have in comparison to their classmates.

Described modes, training and competition, will be available only for students, statistic however can be also viewed by teacher and system administrator with the addition that they can see statistics of all students.

Both of those roles can also view, add, edit and delete questions. Moreover, administrator can also view users, as well as add, edit and delete them, while teacher does not have this functionality.

2.2. Positioning

2.2.1. Problem statement

My application helps solve the problem of interesting children to learn mathematic and overcome their lack of motivation. This problem affects students from class 4 to 6 in primary school and their teachers. It causes worse results in studying, lack of basic knowledge among students and extended learning process. A successful solution would help children with learning and increase their motivation at the same time allowing teachers check students' knowledge of specified topic.

2.2.2. Product position statement

This application is destined for teachers who want a new study aid that will help them in teaching. It is a web-based software system that allows students to check and increase their knowledge using questions prepared by the teacher. Unlike other educational games my application combines training process with competition, where kids can test their newly gained knowledge with their classmates.

2.3. User summary

Table 2.1 User Summary

Name	Description	Functionalities
Student	A person in age from 10 to 13 attending class from 4 to 6 in primary school.	• Can register to the system. • Can take part in the quiz in mode training. • Can take part in the quiz in mode competition. • Can see statistics of students from the same class.
Teacher	A person who teaches mathematic in primary school.	• Can view statistics of all users. • Can view questions. • Can add questions. • Can edit questions. • Can delete questions.
System Administrator	One person chosen among teachers.	• Can view statistics of all users. • Can view questions. • Can add questions. • Can edit questions. • Can delete questions. • Can view all users. • Can add new student. • Can add new teacher. • Can edit student. • Can edit teacher. • Can delete student. • Can delete teacher.

2.4. Gamification

2.4.1. Theory

Gamification is still not a very popular term. However, though it was not defined for a very long time it does not mean it was not applied. But what the gamification really is? According to Gabe Zichermann and Christopher Cunningham: *"The process of game-thinking and game mechanics to engage users and solve problems."* [19].

Because we are designing games for players, we need to understand their motivations. To do so, first we need to know, what kinds of players exists. Looking at figure 2.1 we can see that according to Bartle, four types can be distinguished.

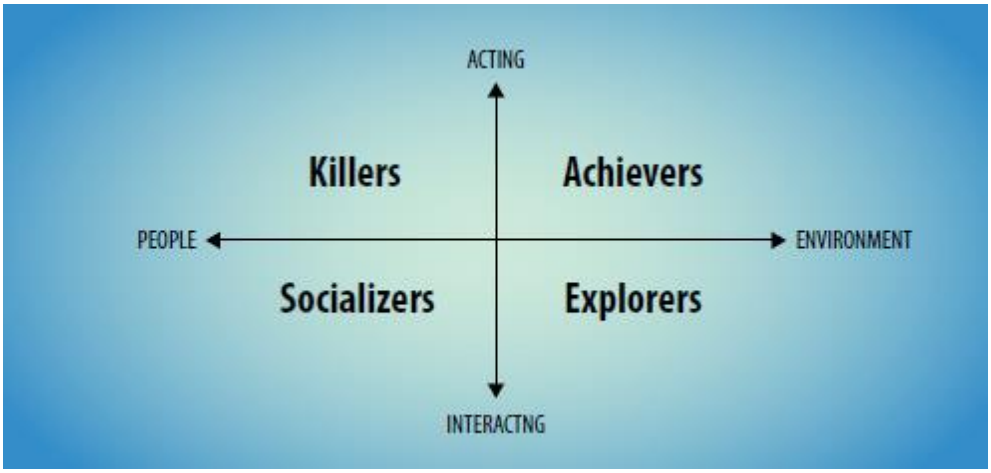

Fig. 2.1 Bartle's player types (source: [19]).

- Achievers – they are focused on collecting everything that can be collected in a game, for example badges, achievements and points.
- Explorers – their goal is to know the whole game, they like discovering new things and finding hidden surprises.
- Socializers – this group of players needs an interaction with other users. A game is a tool for them.
- Killers – they want to win in a way that someone needs to lose.

For each of those players different methods of engaging will be more appealing. Those engaging method can be called Game Mechanics. In the book "Gamification by design" we can find following list of game mechanisms [19]:

- Points,
- Levels,
- Leaderboards,
- Badges,
- Onboarding,
- Challenges/Quests,
- Engagement Loop.

Points are the core of a gamification in an application. It is hard to imagine a game without them. They serve a variety of purposes. There are five main point designs. They can be used separately or combined into larger designs. Those basic designs are:

- Experience points, which indicates the rank of a player. User cannot lose them, only can earn more of them.
- Redeemable points, like virtual currency. Can be trade for some goods. If users will be not satisfied with what they can receive in exchange for those points, they will lose their interest.
- Skill points, which user can earn doing specified activities in the system.
- Karma points, which players can share among each other's. Users have no direct benefit for them.

- Reputation points, used when trust between players is required.

Levels indicate progress. Sometimes they are a game key elements and sometimes they are passive. Their difficulty is not linear, figure 2.2 illustrates example of the progression. The goal is to find the golden mean, so that the levels were not too easy – users would become quickly bored, and not too difficult, because they may become discouraged.

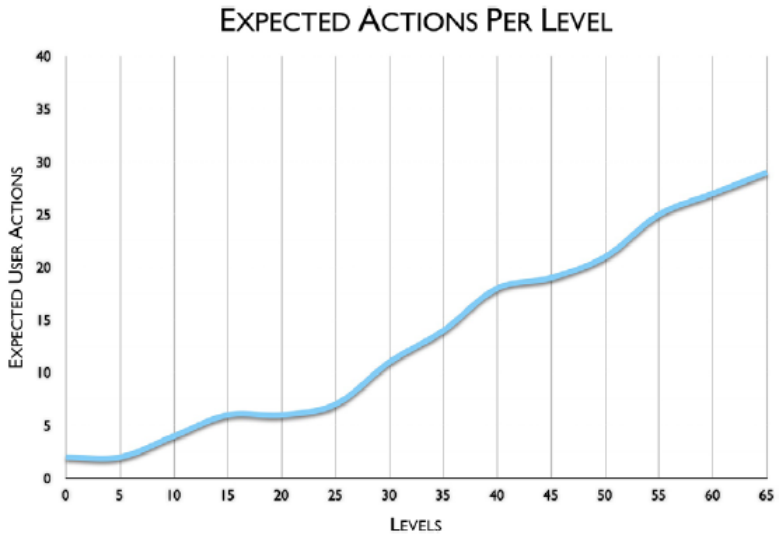

Fig. 2.2 Level complexity (source: [19]).

Together with levels progress bars can be used. They indicate how close a user is to complete something.

Leaderboards can be divided into two categories: no-disincentive leaderboards and infinite leaderboards. The no-disincentive leaderboard shows player in the middle, allowing him to see how many points he needs to increase his position and how close is the next user. The infinite leaderboard displays all users, in some cases allowing a player to filter the results, for example to display only friends.

Badges can be appealing for users who like to collect things, or for those who feel the excitement when a new badge appears unexpectedly. Sometimes they can replace levels.

Onboarding is a process of familiarizing a new user with the system. Those couple of minutes after a player launches the application for the first time are crucial. That is when he makes a lot of decisions concerning the system.

Challenges and quests makes a system more intrigue. A player should always be able to take part in a challenge, matched to his level.

Social engagement loops helps a designer think about how user plays with the system, when he ends and why he is coming back. Motivating emotion (like looking for fun, curiosity) causes social call to action (for example competition) which leads to user re-engagement (using games, challenges) which is followed by visible progress / reward (earning points and seeing statistics among others). That loops back to motivation emotion (Fig. 2.3).

Fig. 2.3 Social engagement loop (source: [19]).

All those examples of gamification can be found in many applications from different areas, like marketing, health, work and education. Gamification has a great potential and correctly used can help to create an amazing systems that will wonderfully serve users.

2.4.2. Usage in application

In my application I decided to use experience points. Users will earn them in competition mode. Earning specified number of those points will cause entering new level. On each level user will receive two types of bonuses. Depending on what level the player is, different number of bonuses he will get. They will be added linearly, so on level 1, after registration, user will have one bonus of each type, on level 2 two more of each type will be added, and so on. Mentioned bonuses are:

- Half-Half Bonus (Pół na pół),
- More Time Bonus (Więcej czasu).

Half-Half Bonus causes that two answers are dropped and only two – one correct and one incorrect – will stay.

More Time Bonus gives an additional minute to a competition time for a player who used it, so from 5 minutes time for this user is extended to 6 minutes.

Bonuses can be used only in competition mode. Tooltips explaining the possibility of each bonus will be provided. During competition Half-Half Bonus for one question can be selected only once, while More Time Bonus can be used many times. After using one of the bonuses, its number decreases. Their role is to increase the chances of player, who will use them, to win. The idea is that players will want to enter next level in order to receive more bonuses.

After registration, user will have available only one category. Reaching certain level will cause that new categories will get unblocked. Playing one topic over and over again would get boring. New categories together with number of level necessary to unblock them will be displayed, so that players have something to aim at.

Moreover, the infinite leaderboard will be provided, with the restriction that student can see scores only from players from the same class he is in. Also, in training mode progress bar will be introduced, so that player knows how close he is for achieving the goal.

2.5. Existing solutions

In the Internet many educational online games can be found. Below I will try to describe some of them, together with pointing their advantages and disadvantages.

2.5.1. Learning Apps

First of selected portals is Learning Apps. It is German portal, however it is available in 20 languages, including Polish. It offers multiple number of blocks, called apps, which are divided into categories as Mathematics, History, Chemistry and many others. Those categories are later divided into subcategories, so that users in easy way can find topic they want to practice (Fig. 2.4).

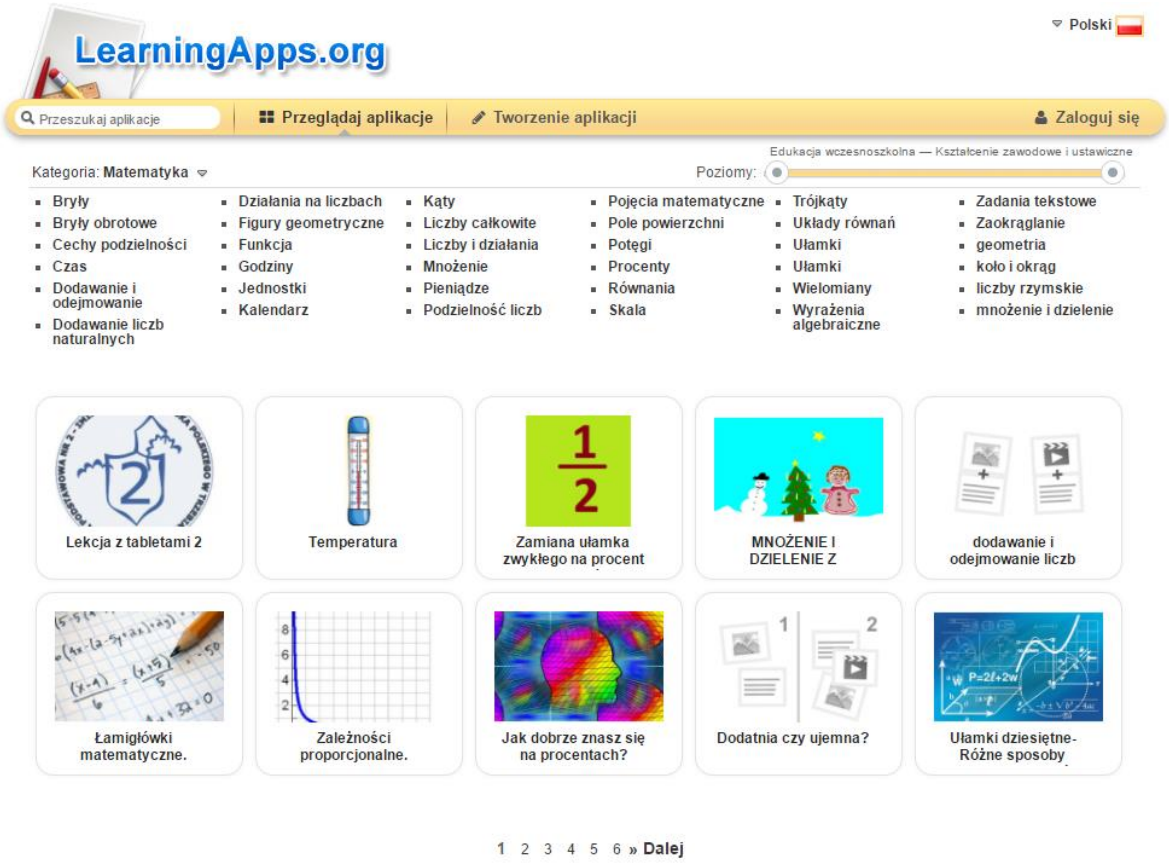

Fig. 2.4 Learning Apps - Examples of applications in category Mathematic (source: [12]).

Each app has one of many possible forms, for example Matching Pairs or Multiple-Choice Quiz. To play them you do not have to create an account.

What is interesting about this portal is that those apps can be created by anyone. You can start creating your own block without being logged in, but to save it you need an account. Later you

can set it to be a private application, where you receive a link to this game that you can pass to your friends, or a public one, where everyone can play it. Figure 2.5 shows example of application that I created.

Fig. 2.5 Learning Apps - Example of created application (source: [12]).

Advantages:

- Applications in Polish language are available.
- No need to create an account to play the games.
- A great amount of categories.
- A lot of already prepared, ready to play games.
- Possibility to create your own game.
- Various templates to choose from.

Disadvantages:

- Because those applications can by create by anyone, quality of some of them is very poor. To help with that, before choosing a game its ratings are shown as well as number of people who played it. However, it can happen that in a game will be mistakes and kids that are playing it will learn something that is not true.
- No gamification and continuity of playing.

2.5.2. Khan Academy

Next application is called Khan Academy. It is non-profit company, which was created to *"[...] provide a free, world-class education to anyone, anywhere."* [11] Was founded in 2008 by educator Salman Khan. On this portal you can find many categories and subcategories. Each of them offers video tutorials that explains selected topic as well as exercises to check if knowledge from those tutorials was fully acquired (Fig. 2.6).

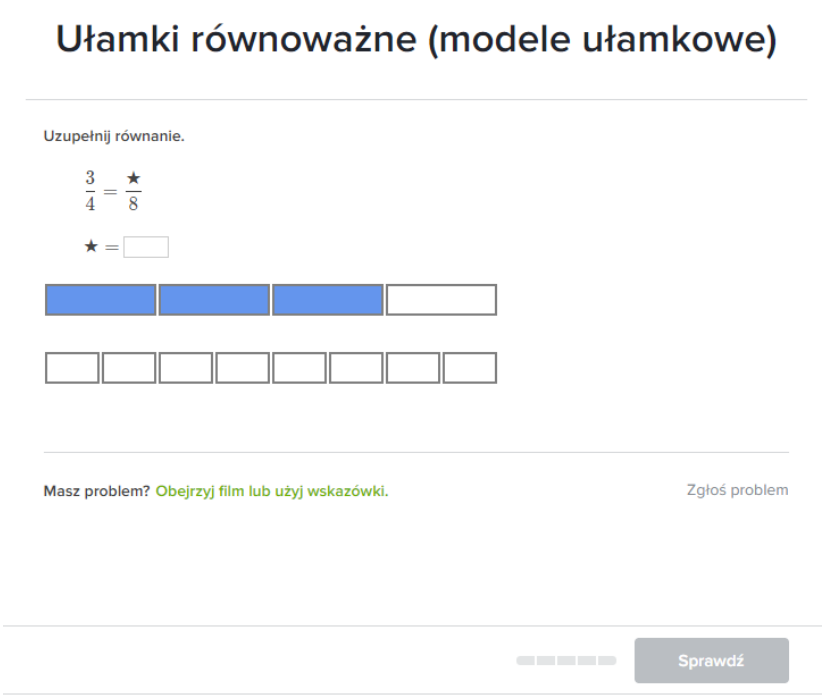

Fig. 2.6 Khan Academy - Exercise after video tutorial (source: [11]).

Apart from that it also allows users to enter missions, which aim is for kid to gain specified skills. As presented in figure 2.7, gamification method were here applied – playing missions and mastering topics unblocks new avatars and badges, that can later be seen in a user profile.

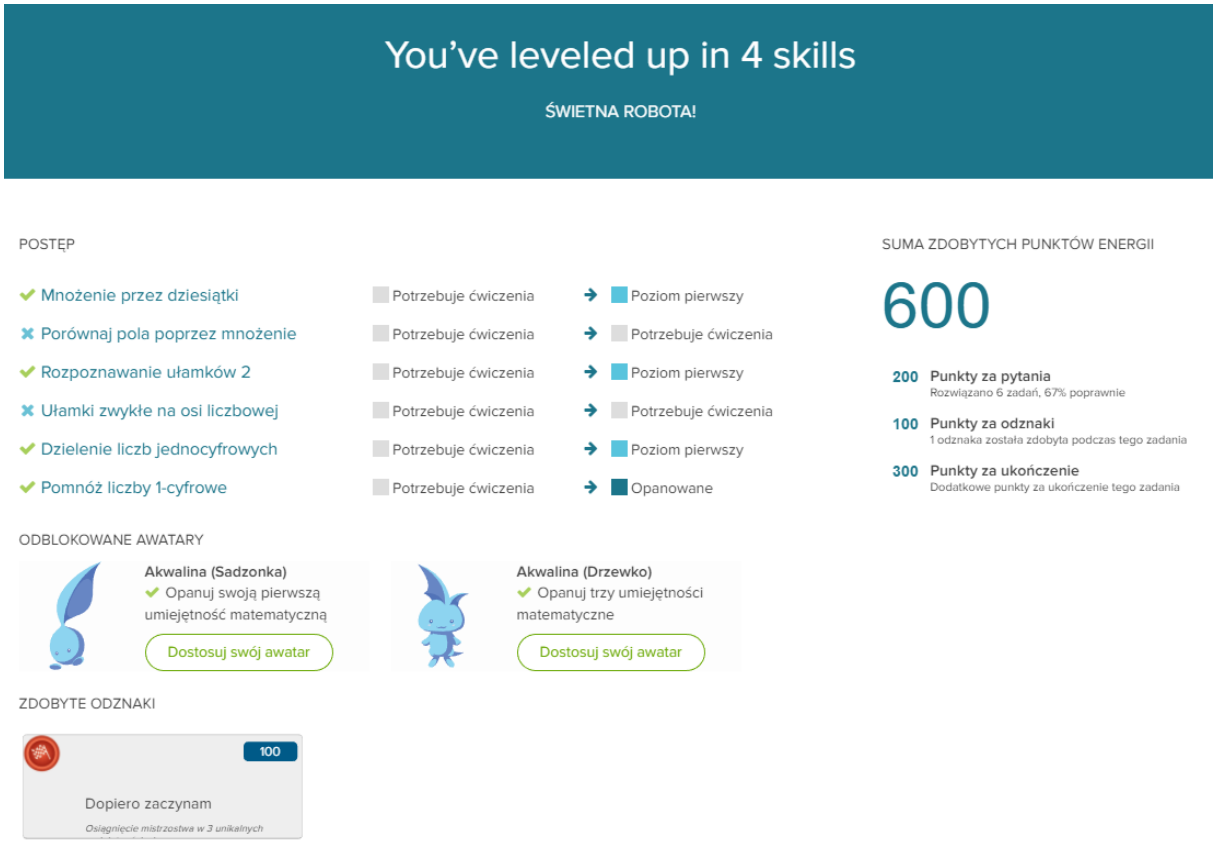

Fig. 2.7 Khan Academy - Popup after finish of exercise in mission (source: [11]).

Advantages:

- A lot of subjects to choose from.
- Video tutorials explaining the topic. If student did not understood the topic in a class, he can watch explanation at home, as many times as he needs.
- To watch tutorials account is not needed.
- There exists Polish version of the portal.
- Gamification elements applied.
- Content is prepared by qualified persons.

Disadvantages:

- Not all texts and labels are translated into Polish (Fig. 2.7).
- No possibility to compete with other users.

2.5.3. Kahoot!

Kahoot! is a typical quiz-based game, launched in Norway, in 2013. It is great for playing in the classroom, because can engage all students at once.

To fully exploit the potential of this application, first, person needs to create an account. Then he has the possibility to create his own quiz, setting questions and answers as well as time limit for each question. Then, when the quiz is ready, after choosing a mode – Player vs. Player or Team vs. Team, and setting game options, such as if order of questions and answers should be randomized (Fig. 2.8), he receives a unique pin.

Fig. 2.8 Kahoot! - Preview of created quiz before start (source: [10]).

Each user (or team) who want to take part in the competition need his own device. Each of them enters Kahoot! page and puts generated pin. This way, he enters the game. Questions are displayed on one, main device. Players on theirs see only four rectangles with different colors and symbols (Fig. 2.9).

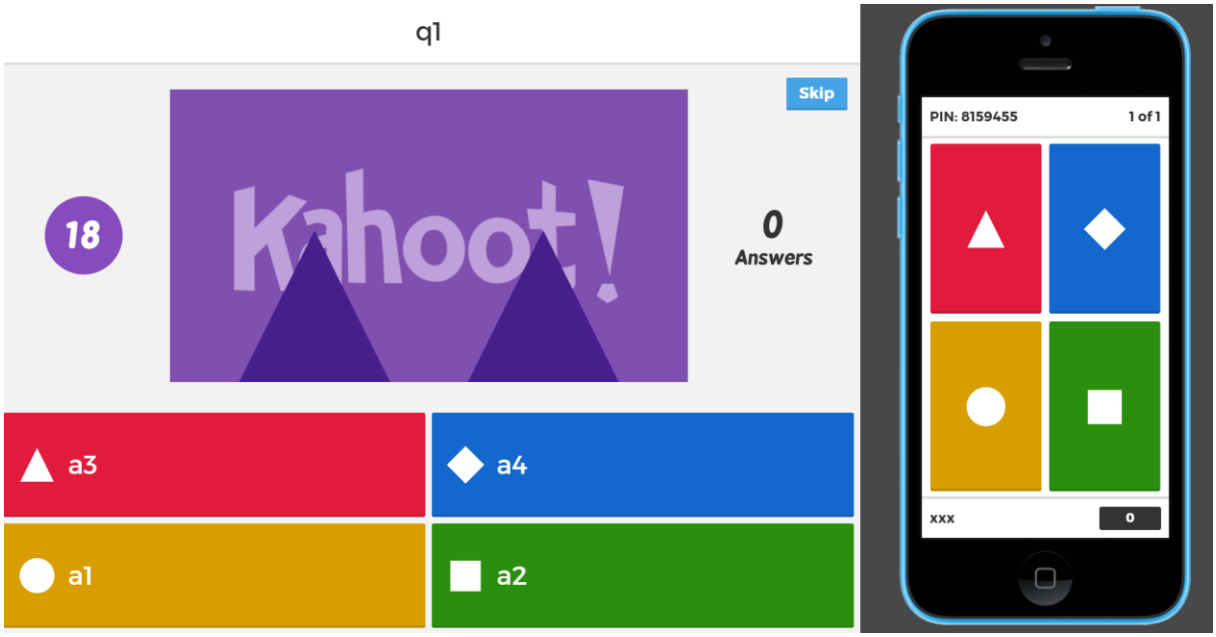

Fig. 2.9 Kahoot! - Preview of created quiz during the game (source: [10]).

Advantages:

- Simple, but very appealing interface.
- Possibility to see leader board.
- After every question number of players who selected specified answer is shown. This can help teacher to see were kids have problem.
- Wide game configuration is possible.
- It creates healthy competition among students.

Disadvantages:

- No Polish version. Quiz can be created in any language, but in order to create it, at least basic knowledge of English is necessary.
- Small difference between modes. Team vs. Team mode extends Player vs. Player mode by adding 5 seconds of "Team talk" before starting counting down the time set for answering the question.

2.5.4. Duolingo

Another application I selected is Duolingo. It focuses strictly on helping to learn a language. As they say on their webpage: *"It's hard to stay motivated when learning online, so we made Duolingo so fun that people would prefer picking up new skills over playing a game."* [6]. Depending on what language you know, different courses are available to you.

Learning is divided into modules, like Food, Animals or Time. Each module takes specific place in ladder and to unblock new stage, you need to pass all modules in the stage before (Fig. 2.10). Passing a module is done by answering on a set of prepared questions. Those questions come in various forms, such as writing heard text and selecting an image presenting displayed text.

Fig. 2.10 Duolingo - Ladder modules (source: [6]).

In Duolingo they think that regularity is very important, so they recommend to play every day, at least for a little while. To motivate users to do so, you can earn lingots – virtual money, for playing seven days in a row, as well as for reaching next level. Lingots can be traded for power-ups, practice bonuses and bonus skills (Fig. 2.11).

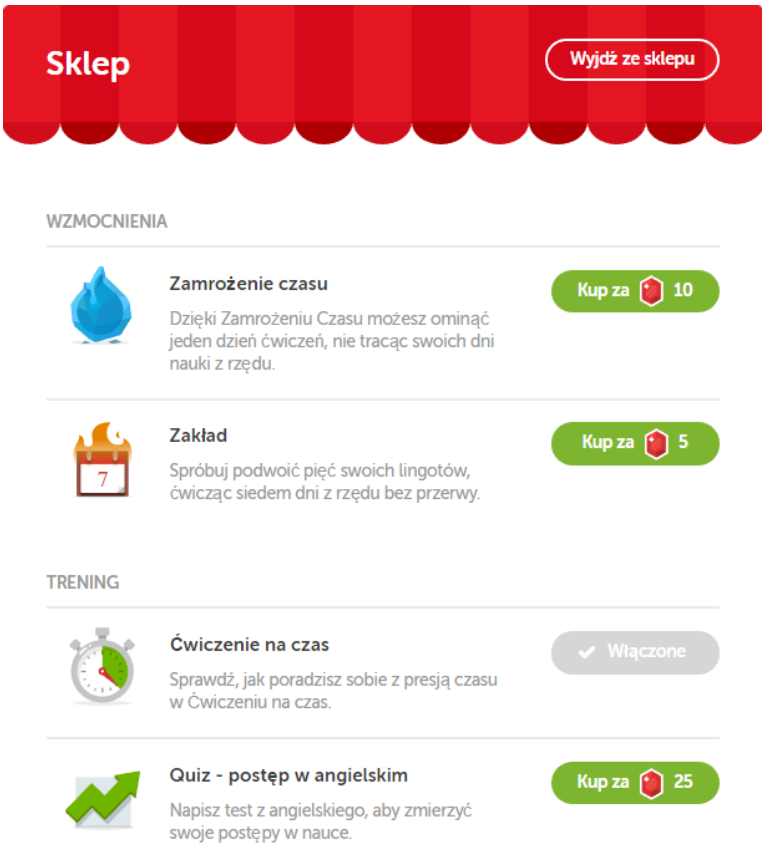

Fig. 2.11 Duolingo - Shop (source: [6]).

Advantages:

- Platform includes website and a mobile application.
- Good training system, where questions for which user selected wrong answer are asked again.
- Progress bar that shows how far you are from completing a training.
- Wide collection of questions types.
- Gamifications elements – virtual currency with store where it can be spent, unblocking new modules with progress in the game.

Disadvantages:

- No possibility to compete with other users.

2.5.5. Squla

The last application is Squla. It is a platform available as a webpage and in a form of a mobile application. It offers thousands of exercises in several categories, in form of quizzes, missions and games with different difficulties (Fig. 2.12). Moreover, educational videos are available. It is intended for kids in pre-school age and for primary school pupils. [18]

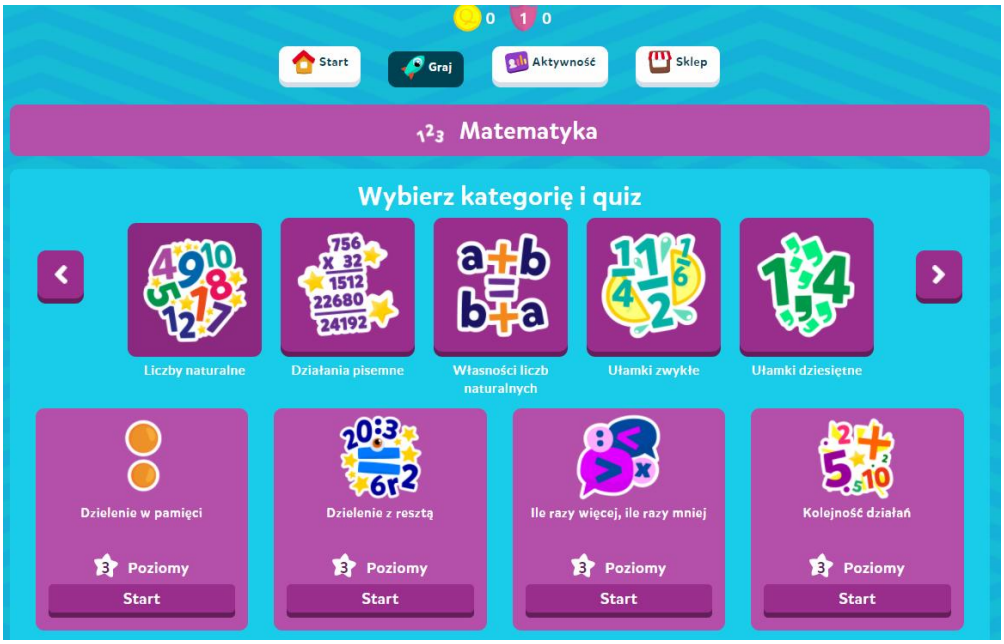

Fig. 2.12 Squla – Different categories (source: [18]).

Number of types of questions in a quiz is very wide and for every good answer user receives virtual coins, that later can be exchange in Squla shop for real gifts (Fig. 2.13).

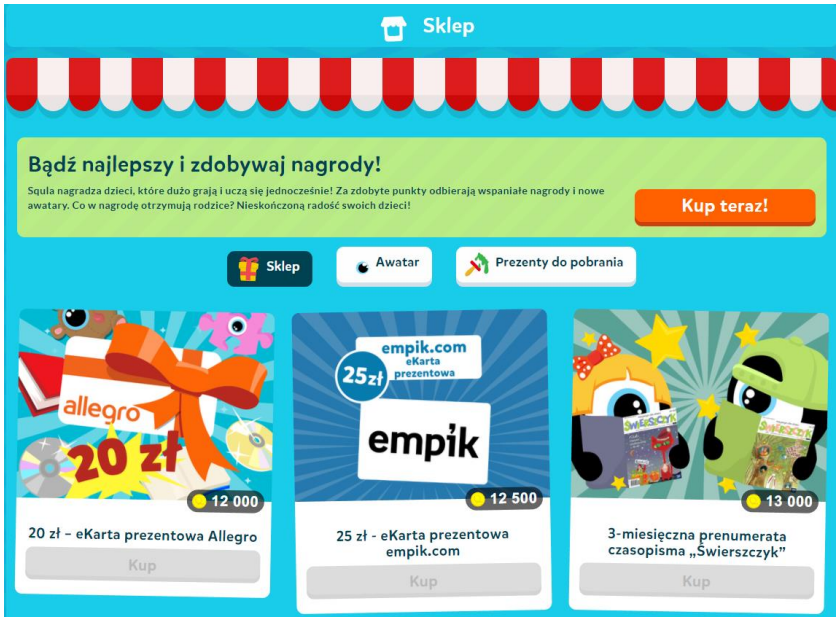

Fig. 2.13 Squla – Shop (source: [18]).

Advantages:

- Version of portal in Polish.
- All tasks are consistent with the core curriculum of Polish Ministry of Education.
- A wide range of subjects, types of questions and modes.
- Virtual currency that can be trade into physical gifts.
- Possibility to see earned achievements and scores.
- User can train by himself, or compete with a friend.

Disadvantages:

- To be able to get access to all game resources, payment is necessary. Without it, only couple of quizzes are available, in order for user to see the potential of the platform.

2.5.6. Comparison between

I want my application to connect the best features of all mentioned applications. To do it, I decided that:

- Game will be available in Polish language.
- Different categories of questions to choose from.
- Questions will be approved by a qualified person.
- Possibility to add more questions.
- Training, where questions for which user selected wrong answer are asked again.
- Competition, to introduce a healthy competition between users.

3. Application design

3.1. Requirement specification

3.1.1. Functional requirements

General:

- User shall log in to the system.
- User shall log out from the system.
- Student shall register into the system.
- System shall validate if all fields are correctly filled during registration.

Training:

- Student shall choose category for training.
- Depending on Student level, different categories shall be available.
- Student shall choose number of questions for training.
- Student shall take part in training.

Competition:

- Student shall choose category for competition.
- Depending on Student level, different categories shall be available.
- Student shall take part in competition.

Statistics:

- Student shall view statistics only of users from the same class as Student.
- System administrator and Teacher shall view statistics of all Students.
- For Student, each row in the statistics shall consist of first name, last name, level and points.
- For System administrator and Teacher, each row in the statistics shall consist of first name, last name, level, points and class.
- Statistics shall include paging.
- For System administrator and Teacher, statistics shall include filtering by class.
- User shall sort statistics by all columns.
- By default, statistics shall be ordered firstly by points, then by last name, then by first name.
- User shall search through the statistics.

List of questions:

- System administrator and Teacher shall view list of all questions.
- Each row in the list of questions shall consist of Id, category, question, four answers and correct answer.
- List of questions shall include paging.
- List of questions shall include filtering by category.
- System administrator and Teacher shall sort list of questions by all columns.
- By default, list of questions shall be ordered by Id.
- System administrator and Teacher shall search through the list of questions.
- System administrator and Teacher shall add new question.
- System administrator and Teacher shall edit a question.

- System shall validate if all fields are correctly filled during adding or editing a question.
- System administrator and Teacher shall delete a question.

List of users:

- System administrator shall view list of all users.
- Each row in the list of users shall consist of user role, last name, first name and login.
- List of users shall include paging.
- List of users shall include filtering by role.
- System administrator shall sort list of users by all columns.
- By default, list of users shall be ordered firstly by role, then by last name, then by first name.
- System administrator shall search through the list of users.
- System administrator shall add new Student.
- System administrator shall add new Teacher.
- System administrator shall edit a user.
- System shall validate if all fields are correctly filled during adding or editing a user.
- System administrator shall delete a user with role Student or Teacher.

3.1.2. Non-functional requirements

- System shall be destined for one school.
- Multiple users shall use the system at once.
- Users' passwords shall be secured.
- System shall work on Chrome, Mozilla Firefox, Opera and Microsoft Edge browsers.
- System shall be responsive.

3.2. User interface

When designing my application I had to think also about its interface. It is very important part of a system as it is used to interact with the user.

3.2.1. Responsive web design

Currently we have variety of devices that can be used to load websites. As we can see on the figure below (Fig. 3.1), in 2017 in almost all studied countries, majority of people have both desktop computers and mobile phones [4]. So, when adapting an application only to desktop, we will lose a lot of users. This is where Responsive Web Design comes to the rescue.

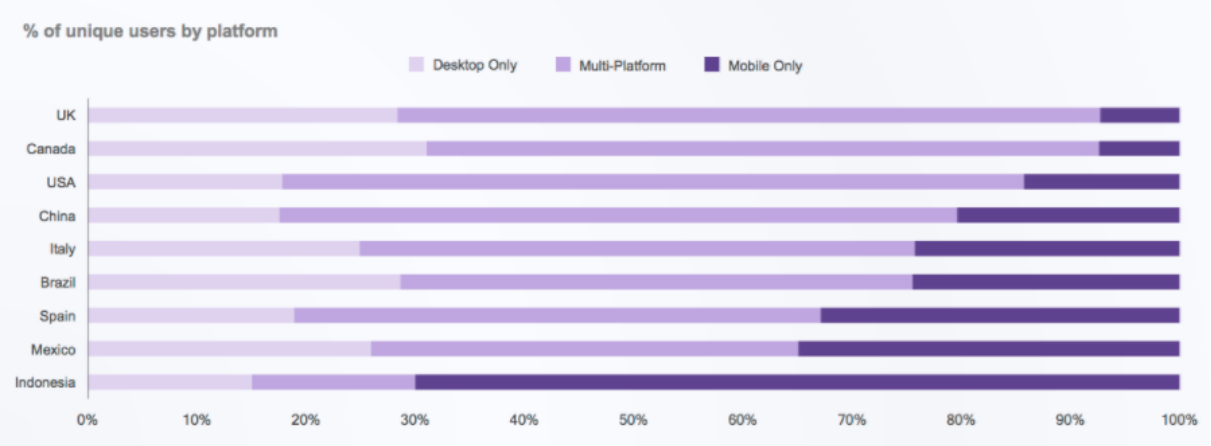

Fig. 3.1 Multi-platform device use 2017 (source: [4]).

Responsive Web Design is a way of designing a website to look good on any device it is opened on. Webpage content will adapt to the size of the device's screen. This approach allows creating applications that will be pleasing for all users, no matter if they prefer desktop, tablet or mobile phone.

3.2.2. Flat design

The application also needs to be enjoyable when it comes to graphics. Because I'm designing an application destined for kids with age between 10 and 13, I want it to be colorful but not too childish. I decided on a blue to be a main color of application. According to the book "The Principles of Beautiful Web Design" by Jason Beaird, blue color symbolizes intelligence and openness [2]. It also has calming properties.

I also decided on using flat design. It is an approach of using minimalistic graphics, so that nothing will distract from user experience [14]. To do it, elements like gradients and shadows are eliminated. Appling flat design helps to create responsive web design too, as the systems that are using it are more clear and readable.

Images that I use in the application are taken from Freepik [9] and Flaticon [8] portals. Some of them I modified using Gimp 2 software.

3.3. Use-case diagram

To illustrate desired work of the system I prepared Use-case diagram (Fig. 3.2). Because System Administrator has all functionalities of Teacher I defined an inheritance relationship between them. I also applied a generalization of Teacher and Student into User, as they share some of the use cases.

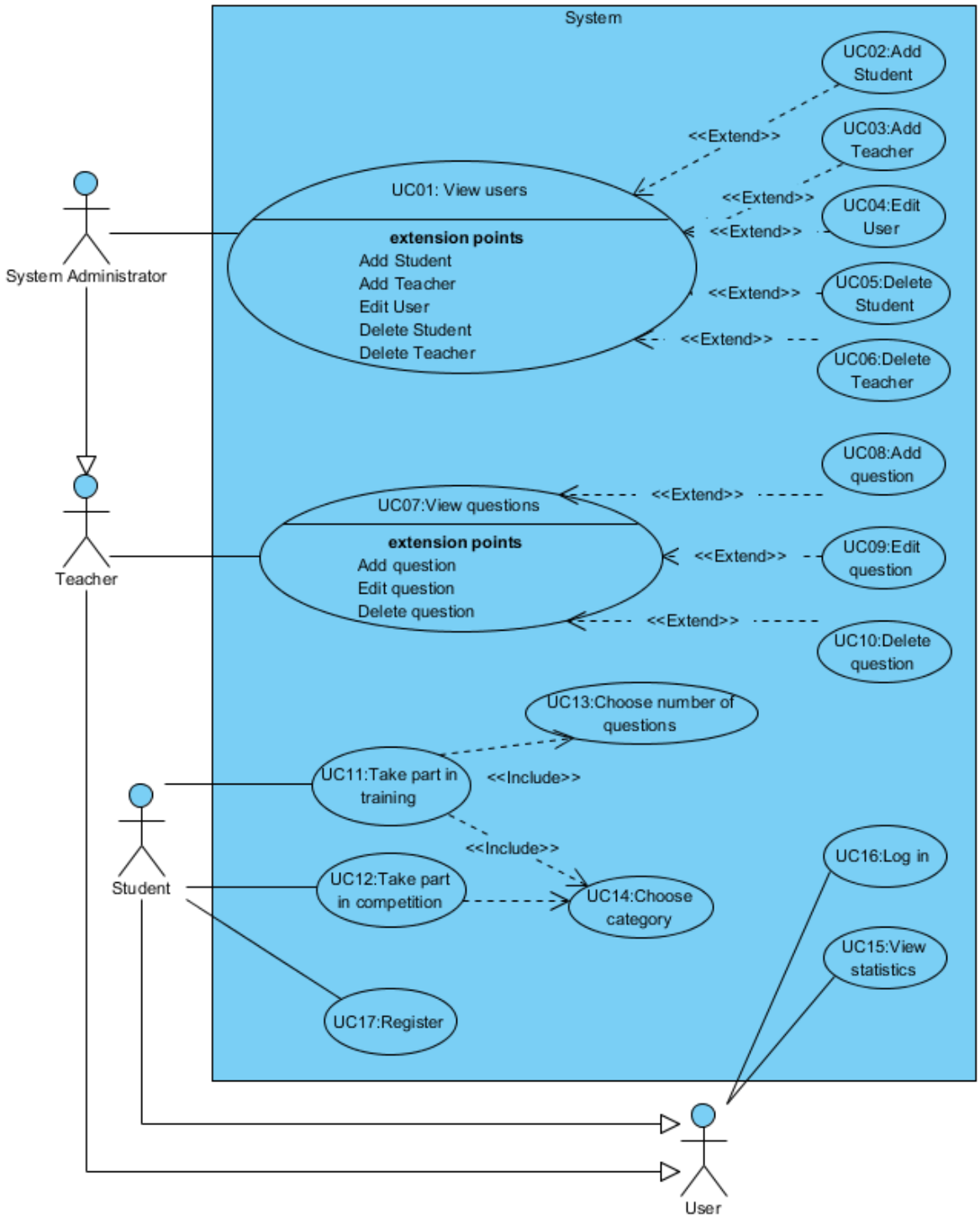

Fig. 3.2 Use-case diagram.

3.4. Use-case specification

Table 3.1 UC01: View users

UC ID:	UC01
Name:	View users
Actors:	System Administrator
Short description:	Viewing list of all users who are registered into the system.
Pre-conditions:	System Administrator is logged in.
Main flow of action:	1. System Administrator clicks on “Wybierz” button under “Zarządzaj użytkownikami” label on the main page. 2. System shows list of users in form of a table. List is sorted firstly by role, then by last name, then by first name.
Alternative flow 1:	System Administrator wants to add Student. 3a. Call use case UC02.
Alternative flow 2:	System Administrator wants to add Teacher. 3a. Call use case UC03.
Alternative flow 3:	System Administrator wants to edit User. 3a. Call use case UC04.
Alternative flow 4:	System Administrator wants to delete Student. 3a. Call use case UC05.
Alternative flow 5:	System Administrator wants to delete Teacher. 3a. Call use case UC06.
Alternative flow 6:	System Administrator wants filter users by role. 3a. System Administrator selects one of available roles from drop-down list located above the table. 3b. System Administrator clicks on “Wybierz” button. 3c. System displays only list of users with selected role.
Alternative flow 7:	System Administrator wants to search for a specific text in a table. 3a. System Administrator enters desired text in a text field with label “Wyszukaj” located above the table. 3b. System displays rows containing specified text.
Alternative flow 8:	System Administrator wants to sort rows by selected column in ascending order. 3a. System Administrator clicks on the header of a column according to which he wants rows to be sorted. 3b. System displays rows sorted according to selected column in ascending order.
Alternative flow 8:	System Administrator wants to sort rows by selected column in descending order. 3a. System Administrator clicks twice on the header of a column according to which he wants rows to be sorted. 3b. System displays rows sorted according to selected column in descending order.

Table 3.2 UC02: Add Student

UC ID:	UC02
Name:	Add Student
Actors:	System Administrator
Short description:	Adding new Student into the system.
Pre-conditions:	System Administrator sees list of users.
Main flow of action:	1. System Administrator clicks on “Dodaj ucznia” button above the table with users. 2. System displays the form for Student registration. 3. Systems Administrator fills the form. 4. System Administrator clicks on “Zapisz” button. 5. System validates entered data. 6. System registers new Student, list of users is displayed.
Alternative flow 1:	System Administrator wants to add more Students. 4a. Systems Administrator clicks on “Dodaj kolejnego” button. 4b. System validates entered data. 4c. System registers new Student, go to step 2.
Alternative flow 2:	System Administrator leaves one or more fields empty. 5a. System informs under empty fields that they must be filled, go to step 3.
Alternative flow 3:	System Administrator enters incorrect number of characters into one or more fields. 5a. System informs under incorrect field what the minimal number of characters is, go to step 3.
Alternative flow 4:	System Administrator enters different data for password and confirm password fields. 5a. System informs that entered passwords do not match, go to step 3.
Alternative flow 5:	System Administrator enters login that already exists in the system. 5a. System informs that user with given login already exists, go to step 3.
Alternative flow 6:	System Administrator cancels adding a Student. 4a. System Administrator clicks on “Anuluj” button. 4b. New Student is not registered into the system, list of users is displayed.

Table 3.3 UC03: Add Teacher

UC ID:	UC03
Name:	Add Teacher
Actors:	System Administrator
Short description:	Adding new Teacher into the system.
Pre-conditions:	System Administrator sees list of users.
Main flow of action:	1. System Administrator clicks on “Dodaj nauczyciela” button above the table with users. 2. System displays the form for Teacher registration. 3. Systems Administrator fills the form. 4. System Administrator clicks on “Zapisz” button.

	5. System validates entered data. 6. System registers new Teacher, list of users is displayed.
Alternative flow 1:	System Administrator wants to add more Students. 4a. Systems Administrator clicks on “Dodaj kolejnego” button. 4b. System validates entered data. 4c. System registers new Teacher, go to step 2.
Alternative flow 2:	System Administrator leaves one or more fields empty. 5a. System informs under empty fields that they must be filled, go to step 3.
Alternative flow 3:	System Administrator enters incorrect number of characters into one or more fields. 5a. System informs under incorrect field what the minimal number of characters is, go to step 3.
Alternative flow 4:	System Administrator enters different data for password and confirm password fields. 5a. System informs that entered passwords do not match, go to step 3.
Alternative flow 5:	System Administrator enters login that already exists in the system. 5a. System informs that user with given login already exists, go to step 3.
Alternative flow 6:	System Administrator cancels adding a Teacher. 4a. System Administrator clicks on “Anuluj” button. 4b. New Teacher is not registered into the system, list of users is displayed.

Table 3.4 UC04: Edit User

UC ID:	UC04
Name:	Edit user
Actors:	System Administrator
Short description:	Editing a User who is registered into the system.
Pre-conditions:	System Administrator sees list of users.
Main flow of action:	1. System Administrator clicks on edit icon in the table with users, in column “Edytuj”, and row with User he wants to edit. 2. System displays the form for editing the User, where all fields are filled with edited User’s data. 3. Systems Administrator changes one or more fields. 4. System Administrator clicks on “Zapisz” button. 5. System validates entered data. 6. System saves User personal data, list of users is displayed.
Alternative flow 1:	System Administrator leaves one or more fields empty. 5a. System informs under empty fields that they must be filled, go to step 3.
Alternative flow 2:	System Administrator enters incorrect number of characters into one or more fields. 5a. System informs under incorrect field what the minimal number of characters is, go to step 3.

Alternative flow 3:	System Administrator enters different data for password and confirm password fields. 5a. System informs that entered passwords do not match, go to step 3.
Alternative flow 4:	System Administrator enters login that already exists in the system. 5a. System informs that user with given login already exists, go to step 3.
Alternative flow 5:	System Administrator cancels editing the User. 4a. System Administrator clicks on “Anuluj” button. 4b. Changed data is not saved, list of users is displayed.

Table 3.5 UC05: Delete Student

UC ID:	UC05
Name:	Delete Student
Actors:	System Administrator
Short description:	Deleting registered Student from the system.
Pre-conditions:	System Administrator sees list of users.
Main flow of action:	1. System Administrator clicks on delete icon in the table with users, in column “Usuń”, and row with Student he wants to delete. 2. System askes for confirmations of deleting the Student. 3. System Administrator clicks “OK”. 4. System deletes the Student, list of users is displayed.
Alternative flow 1:	System Administrator cancels deleting of Student. 3a. System Administrator clicks on “Anuluj” button. 3b. Student is not deleted from the system, list of users is displayed.

Table 3.6 UC06: Delete Teacher

UC ID:	UC06
Name:	Delete Teacher
Actors:	System Administrator
Short description:	Deleting registered Teacher from the system.
Pre-conditions:	System Administrator sees list of users.
Main flow of action:	1. System Administrator clicks on delete icon in the table with users, in column “Usuń”, and row with Teacher he wants to delete. 2. System askes for confirmations of deleting the Teacher. 3. System Administrator clicks “OK”. 4. System deletes the Teacher, list of users is displayed.
Alternative flow 1:	System Administrator cancels deleting of Teacher. 3a. System Administrator clicks on “Anuluj” button. 3b. Teacher is not deleted from the system, list of users is displayed.

Table 3.7 UC07: View questions

UC ID:	UC07
Name:	View users
Actors:	Teacher
Short description:	Viewing list of all questions saved into the system.
Pre-conditions:	Teacher is logged in.
Main flow of action:	1. Teacher clicks on “Wybierz” button under “Zarządzaj pytaniami” label on the main page. 2. System shows list of questions in form of a table. List is sorted by Id.
Alternative flow 1:	Teacher wants to add question. 3a. Call use case UC08.
Alternative flow 2:	Teacher wants to edit question. 3a. Call use case UC09.
Alternative flow 3:	Teacher wants to delete question. 3a. Call use case UC10.
Alternative flow 4:	Teacher wants filter questions by category. 3a. Teacher selects one of available categories from drop-down list located above the table. 3b. Teacher clicks on “Wybierz” button. 3c. System displays only list of questions from selected category.
Alternative flow 5:	Teacher wants to search for a specific text in a table. 3a. Teacher enters desired text in a text field with label “Wyszukaj” located above the table. 3b. System displays rows containing specified text.
Alternative flow 6:	Teacher wants to sort rows by selected column in ascending order. 3a. Teacher clicks on the header of a column according to which he wants rows to be sorted. 3b. System displays rows sorted according to the selected column in ascending order.
Alternative flow 7:	Teacher wants to sort rows by selected column in descending order. 3a. Teacher clicks twice on the header of a column according to which he wants rows to be sorted. 3b. System displays rows sorted according to the selected column in descending order.

Table 3.8 UC08: Add question

UC ID:	UC08
Name:	Add question
Actors:	Teacher
Short description:	Adding new question into the system.
Pre-conditions:	Teacher sees list of questions.
Main flow of action:	1. Teacher clicks on “Dodaj pytanie” button above the table with users. 2. System displays the form for adding a question. 3. Teacher fills the form. 4. Teacher clicks on “Zapisz” button.

	5. System validates entered data. 6. System saves new question, list of questions is displayed.
Alternative flow 1:	Teacher wants to add more questions. 4a. Teacher clicks on “Dodaj kolejne” button. 4b. System validates entered data. 4c. System saves new question, go to step 2.
Alternative flow 2:	Teacher leaves one or more fields empty. 5a. System informs under empty fields that they must be filled, go to step 3.
Alternative flow 3:	Teacher enters incorrect number of characters into one or more fields. 5a. System informs under incorrect field what the minimal number of characters is, go to step 3.
Alternative flow 4:	Teacher cancels adding a question. 4a. Teacher clicks on “Anuluj” button. 4b. New question is not added into the system, list of questions is displayed.

Table 3.9 UC09: Edit question

UC ID:	UC09
Name:	Edit question
Actors:	Teacher
Short description:	Editing a question previously saved into the system.
Pre-conditions:	Teacher sees list of questions.
Main flow of action:	1. Teacher clicks on edit icon in the table with questions, in column “Edytuj”, and row with question he wants to edit. 2. System displays the form for editing the question, where all fields are filled with edited question’s data. 3. Teacher changes one or more fields. 4. Teacher clicks on “Zapisz” button. 5. System validates entered data. 6. System saves question data, list of questions is displayed.
Alternative flow 1:	Teacher leaves one or more fields empty. 5a. System informs under empty fields that they must be filled, go to step 3.
Alternative flow 2:	Teacher enters incorrect number of characters into one or more fields. 5a. System informs under incorrect field what the minimal number of characters is, go to step 3.
Alternative flow 3:	Teacher cancels editing the question. 4a. Teacher clicks on “Anuluj” button. 4b. Changed data is not saved, list of questions is displayed.

Table 3.10 UC10: Delete question

UC ID:	UC10
Name:	Delete question
Actors:	Teacher
Short description:	Deleting a question previously saved into the system
Pre-conditions:	Teacher sees list of questions
Main flow of action:	1. Teacher clicks on delete icon in the table with questions, in column "Usuń", and row with question he wants to delete. 2. System askes for confirmations of deleting the question. 3. Teacher clicks "OK". 4. System deletes the question, list of questions is displayed.
Alternative flow 1:	Teacher cancels deleting of question. 3a. Teacher clicks on "Anuluj" button. 3b. Question is not deleted from the system, list of questions is displayed.

Table 3.11 UC11: Take part in training

UC ID:	UC11
Name:	Take part in training
Actors:	Student
Short description:	Playing in the training mode.
Pre-conditions:	Student is logged in.
Main flow of action:	1. Student clicks on "Wybierz" button under "Trening" label on the main page. 2. System displays categories to choose for training (UC14). 3. System displays filed to enter number of questions (UC13). 4. System starts a training. List of questions, based on selected category and number of questions, is prepared. 5. Student click on "Start" button. 6. System displays question and four answers. 7. Student selects correct answer. 8. Student clicks on "Koniec" button. 9. System validates that the answer was correct. 10. Training is completed, for all questions correct answer was chosen.
Alternative flow 1:	Student selects incorrect answer. 7a. Student selects incorrect answer. 7b. Student clicks on "Koniec" button. 7c. System validates that the answer was incorrect, go to step 6.
Alternative flow 2:	Not all questions were asked or not for all correct answer was selected. 8a. Student clicks on "Następne pytanie" button 8b. System validates that the answer was correct, go to step 6.

Table 3.12 UC12: Take part in competition

UC ID:	UC12
Name:	Take part in competition
Actors:	Student
Short description:	Playing in the competition mode.
Pre-conditions:	Student is logged in.
Main flow of action:	1. Student clicks on “Wybierz” button under “Pojedynek” label on the main page. 2. System displays categories to choose for competition (UC14). 3. Second player enters his login and password. 4. Second player clicks on “Kontynuuj” button. 5. System validates second player login and password. 6. System starts a competition. List of questions, based on selected category, is prepared. 7. First player clicks on “Start” button. 8. System displays question and four answers. 9. First player selects correct answer. 10. First player clicks on “Koniec” button. 11. System validates that the answer was correct. 12. First player finishes the competition. 13. Second player clicks on “Start” button. 14. System displays question and four answers. 15. Second player selects correct answer. 16. Second player clicks on “Koniec” button. 17. System validates that the answer was correct. 18. Second player finishes the competition. 19. System displays competition results, player in the first place and player in the second place together with points. 20. Student clicks on “Dalej” button. 21. System displays list of questions used in the competition together with correct answers and the answers selected by players.
Alternative flow 1:	Second player leaves one or more fields empty while entering data. 5a. System informs under empty fields that they must be filled, go to step 3.
Alternative flow 2:	Second player enters login that is not registered in the system. 5a. System informs that user with given login does not exist, go to step 3.
Alternative flow 3:	Second player enters incorrect password for given login. 5a. System informs that for given login wrong password is entered, go to step 3.
Alternative flow 4:	Second player enters data of user with other role than Student. 5a. System informs that only Students can take part in competition, go to step 3.
Alternative flow 5:	First player selects incorrect answer. 9a. First player selects incorrect answer. 9b. First player clicks on “Koniec” button. 9c. System validates that the answer was incorrect, go to step 8.

Alternative flow 6:	Not all questions were asked for first player. 10a. First player clicks on “Następne pytanie” button 10b. System validates that the answer was correct, go to step 8.
Alternative flow 7:	Second player selects incorrect answer. 15a. Second player selects incorrect answer. 15b. Second player clicks on “Koniec” button. 15c. System validates that the answer was incorrect, go to step 14.
Alternative flow 8:	Not all questions were asked for second player. 16a. Second player clicks on “Następne pytanie” button 16b. System validates that the answer was correct, go to step 14.
Alternative flow 9:	Competition finishes with a tie. 19a. System displays competition results with information about a tie and with points.

Table 3.13 UC13: Choose number of questions

UC ID:	UC13
Name:	Choose number of questions
Actors:	Student
Short description:	Choosing number of questions that will be asked during training.
Pre-conditions:	Student selected training mode.
Main flow of action:	1. Student clicks on “Wybierz” button under selected category label (UC14). 2. System displays filed to enter number of questions. 3. Student enters number of questions. 4. Student clicks on “Start” button. 5. System validates entered data. 6. System starts a training.
Alternative flow 1:	Student leaves the field empty. 4a. System informs under empty field that it must be filled, go to step 3.
Alternative flow 2:	Student enters incorrect number of questions. 4a. System informs about the correct range from which to select a number, go to step 3.

Table 3.14 UC14: Choose category

UC ID:	UC14
Name:	Choose category
Actors:	Student
Short description:	Choosing category to play in training or competition mode.
Pre-conditions:	Student is logged in.
Main flow of action:	1. Student clicks on “Wybierz” button under “Trening” or “Pojedynek” label.

	2. System displays all categories defined in the system. Logos of categories that Student does not have access to are grayed out. 3. Student clicks on “Wybierz” button under selected category label. 4. Category is selected, filed to enter number of questions is displayed.
Alternative flow 1:	Student hovers over grayed out category. 2a. Number of level necessary to unlock this category is displayed, go to step 2.
Alternative flow 2:	Student clicks on the “Wybierz” button under unavailable category. 3a. No action is taken, go to step 2.

Table 3.15 UC15: View statistics

UC ID:	UC15
Name:	View users
Actors:	User
Short description:	Viewing statistics of users.
Pre-conditions:	User is logged in.
Main flow of action:	1. User clicks on “Statystyki” button on the header or (in case of Teacher) on “Wybierz” button under “Statystyki” label on the main page. 2. System shows statistics in form of a table. List is sorted firstly by points, then by last name, then by first name. 3. System shows statistics in an actor-dependent view. In case of a Student, he sees users from his class. In case of a Teacher, he sees all users.
Alternative flow 1:	Teacher wants filter statistics by class. 3a. Teacher selects one of available classes from drop-down list located above the table. 3b. Teacher clicks on “Wybierz” button. 3c. System displays only list of statistics of users from selected class.
Alternative flow 2:	User wants to search for a specific text in a table. 3a. User enters desired text in a text field with label “Wyszukaj” located above the table. 3b. System displays rows containing specified text.
Alternative flow 3:	User wants to sort rows by selected column in ascending order. 3a. User clicks on the header of a column according to which he wants rows to be sorted. 3b. System displays rows sorted according to the selected column in ascending order.
Alternative flow 4:	User wants to sort rows by selected column in descending order. 3a. User clicks twice on the header of a column according to which he wants rows to be sorted. 3b. System displays rows sorted according to the selected column in descending order.

Table 3.16 UC16: Log in

UC ID:	UC016
Name:	Log in
Actors:	User
Short description:	Logging into the system.
Pre-conditions:	User is on the index page.
Main flow of action:	1. User clicks on “Zaloguj się” button. 2. System displays the form for logging. 3. User fills the form. 4. User clicks on “Zaloguj się” button. 5. System validates entered data. 6. System displays main page for User.
Alternative flow 1:	User leaves one or more fields empty. 5a. System informs under empty fields that they must be filled, go to step 3.
Alternative flow 2:	User enters login that is not registered in the system. 5a. System informs that user with given login does not exist, go to step 3.
Alternative flow 3:	User enters incorrect password for given login. 5a. System informs that for given login wrong password is entered, go to step 3.
Alternative flow 4:	User cancels logging. 4a. User clicks on “Powrót” button. 4b.User is not logged into the system, index page is displayed.

Table 3.17 UC17: Register

UC ID:	UC017
Name:	Register
Actors:	Student
Short description:	Registering into the system.
Pre-conditions:	Student is on the index page.
Main flow of action:	1. Student clicks on “Zarejestruj się” button. 2. System displays the form for registering. 3. Student fills the form. 4. Student clicks on “Zarejestruj się” button. 5. System validates entered data. 6. Student is registered into the system, main page is displayed.
Alternative flow 1:	Student leaves one or more fields empty. 5a. System informs under empty fields that they must be filled, go to step 3.
Alternative flow 2:	Student enters incorrect number of characters into one or more fields. 5a. System informs under incorrect field what the minimal number of characters is, go to step 3.
Alternative flow 3:	Student enters different data for password and confirm password fields.

	5a. System informs that entered passwords do not match, go to step 3.
Alternative flow 4:	Student enters login that already exists in the system. 5a. System informs that user with given login already exists, go to step 3.
Alternative flow 5:	Student cancels registering. 4a. Student clicks on “Powrót” button. 4b. Student is not registered into the system, index page is displayed.

3.5. Database schema

Below I present my application’s database schema (Fig.3.3), generated with the use of phpMyAdmin tool.

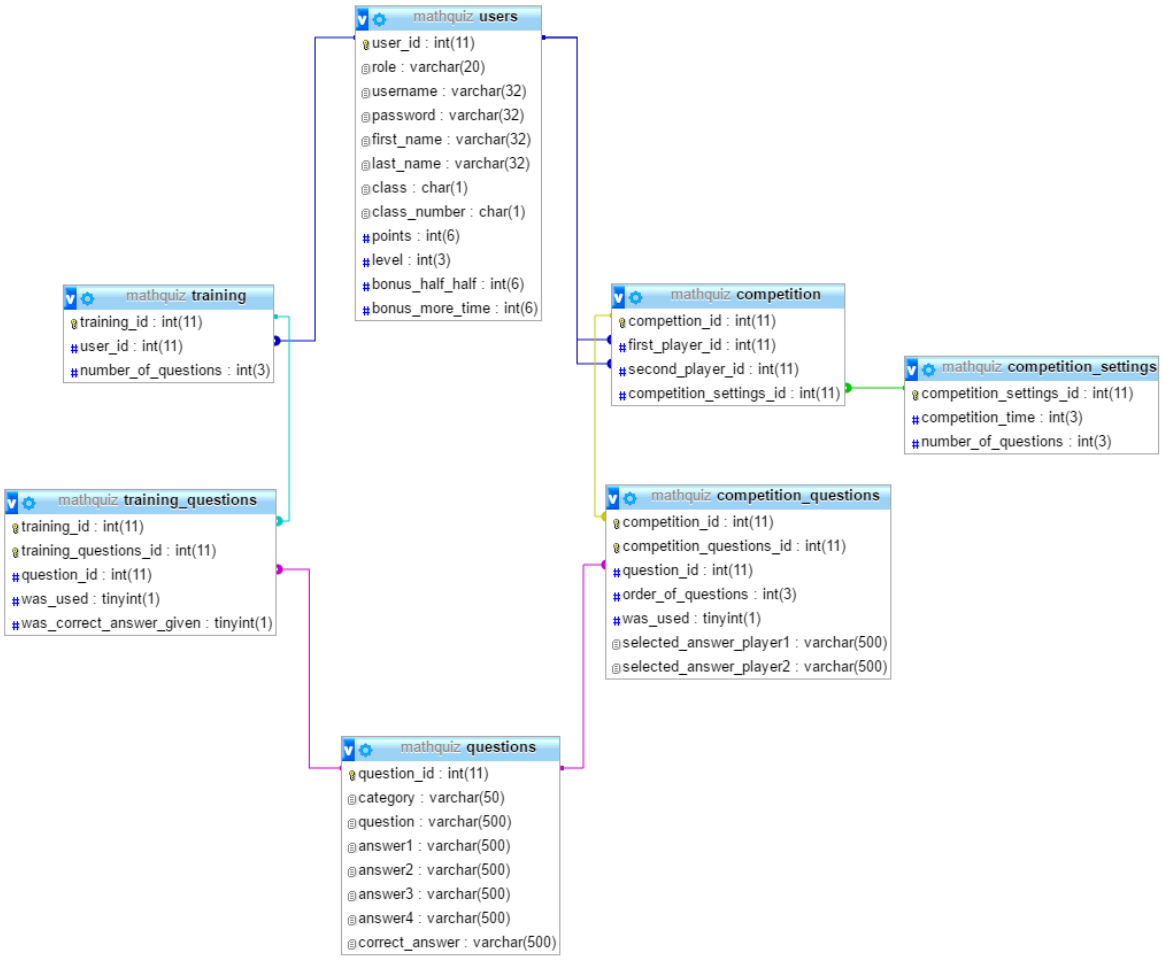

Fig. 3.3 Database schema.

- Table users contains all information about users registered into the system.
- Table questions contains all questions defined in the system.
- Table training contains information about trainings started by Students.

- Table training_questions contains ids of questions selected for the training together with training-specific information.
- Table competition contains information about competitions started by Students.
- Table competition_settings contains information about competition time and number of questions to use in competition.
- Table competition_questions contains ids of questions selected for the competition together with competition-specific information.

4. Implementation

4.1. Relevant technologies

4.1.1. Frontend

Frontend of my application was implemented with the use of HTML5 and CSS3. However, creating an appealing application with the use of only these tools is a hard task. That is why I decided to also use Bootstrap.

Bootstrap is a framework that easies the process of creating a webpage. It has components like navbars and jumbotrons that are adding more attractiveness to the application. It also helps to make a website responsive. To check responsiveness of the system I have used two devices:

- Desktop computer with 24 inches screen and resolution 1920x1080,
- Mobile phone, Samsung Galaxy S7 with 5.1 inches screen and resolution 2560 x 1440.

Figures 4.1 and 4.2 present an index page of my application on the desktop and on the mobile phone respectively. Figures 4.3, 4.4 and 4.5 show question in the training mode on both devices and figure 4.6 displays expanded navbar on the mobile. As we can observe, no scrolling left and right is needed on the small device, however in order to pass the entire content of the pages, sometimes moving up and down is needed.

Fig. 4.1 Index page on desktop computer.

Fig. 4.2 Index page on mobile phone.

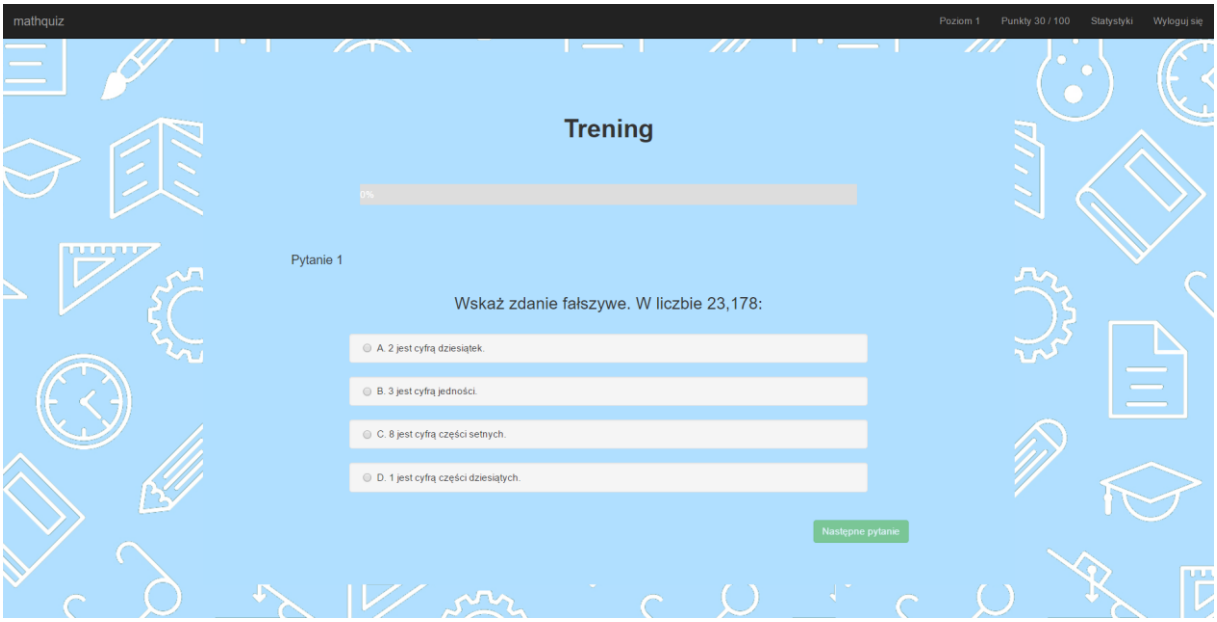

Fig. 4.3 Training on the desktop computer.

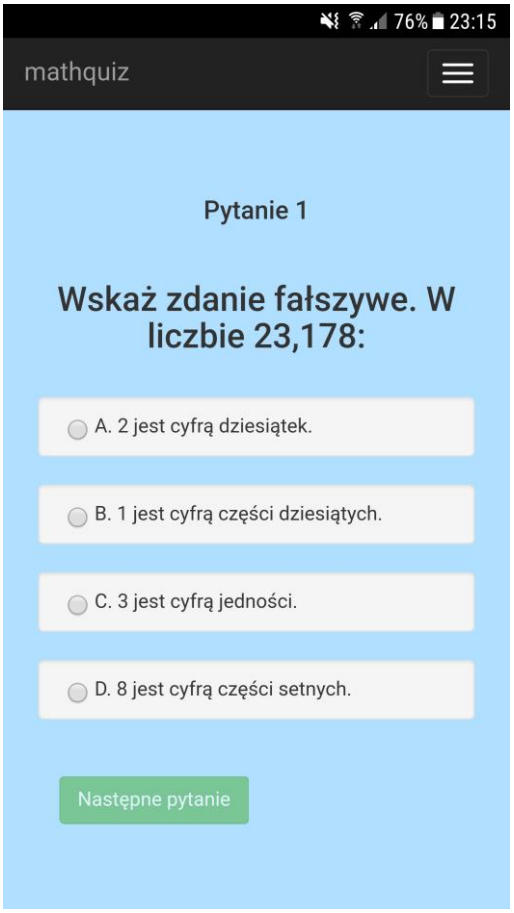

Fig. 4.4 Training on mobile phone – real view.

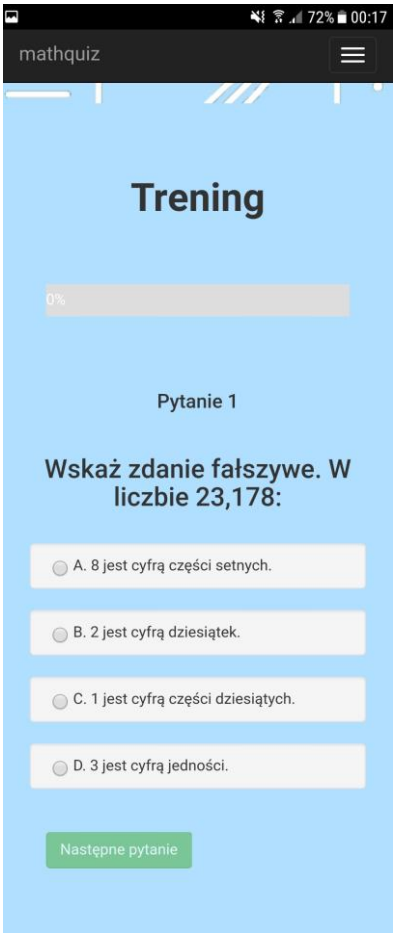

Fig. 4.5 Training on mobile phone – full view.

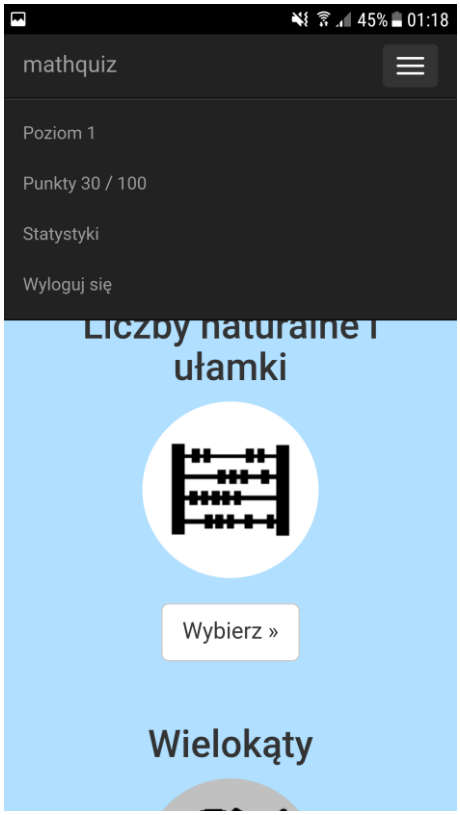

Fig. 4.6 Expanded navbar on mobile phone.

I also used BootstrapValidator [3]. It is a jQuery plugin that helps with form validation. It contains constraints like not empty, minimum and maximum string length and identical, to use for example in fields password and confirm password. Figure 4.7 shows registration page with some error messages.

Fig. 4.7 Fragment of registration page with validation errors.

DataTables is another jQuery plugin [5] that I have used. It helps with displaying the data in the form of a table adding pagination, sorting and searching. The only problem was that all labels were in English, however after some searching I have found that all messages are customizable. Figure 4.8 presents the use of DataTables plugin to display list of users registered in the system.

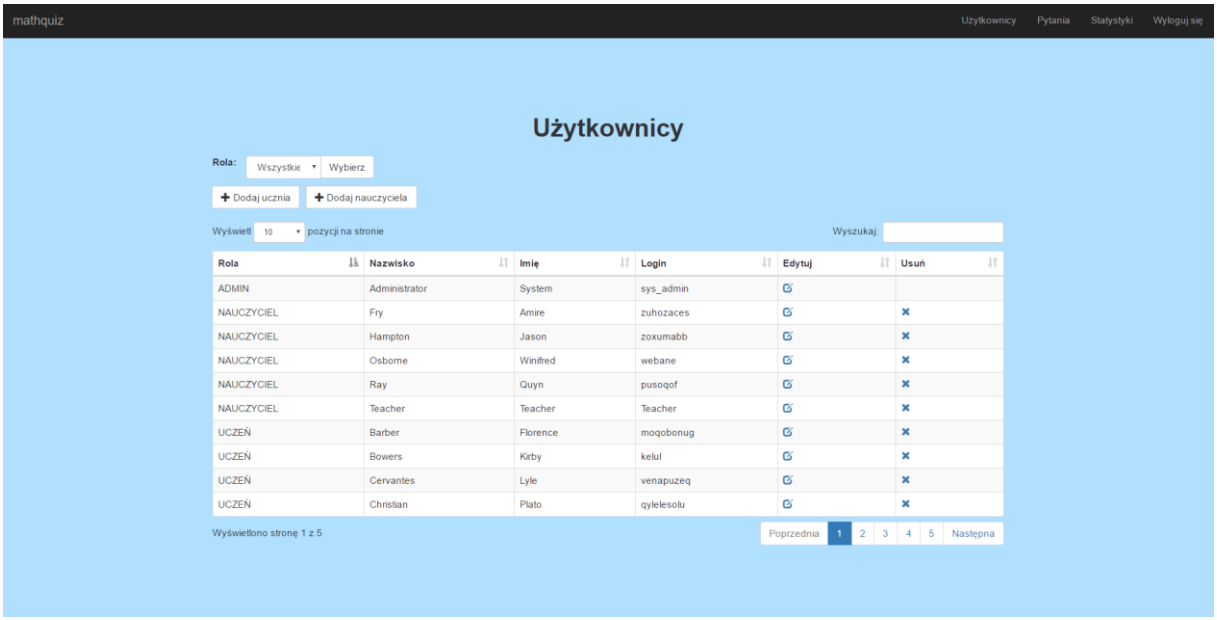

Fig. 4.8 View list of users page.

4.1.2. Backend

For a backend site of my application I decided on PHP. It is a scripting language that is widely used, mostly in web development. PHP stands for *PHP: Hypertext Preprocessor* [16]. It supports many databases, including MySQL. PHP can be used in procedural or object-oriented style. One of the advantages of this language is that it can be used with a very powerful tool which is PhpStorm.

4.1.3. Database

I use XAMPP to set up my local server. It comes with set of tools, MariaDB among them. This database management system is compatible with MySQL, very popular system, which I decided to use in my project. It is XAMPP also provides phpMyAdmin tool, to easily manage the database.

4.2. Mechanism of questions and answers displaying

Questions are the core of my application, so I had to implement their mechanism right. This is why I decided to shuffle the answers for each question. People are mostly visualizers, so this way for Student during his training will not be enough to remember where he clicked last time when he chose a wrong answer, so that he can click on the other answer next time. Instead, he will have to read all the answers again and think about the correct one. Similarly during the competition, second player will have to give the answers thought and not just click on the same places first player did.

4.2.1. Training mode

Figure 4.9 illustrates the mechanism of questions displaying in training mode. The key element here is that answers for which incorrect answer was given are marked as not used at the end of the round, where as a round I mean one turn of asking all of not used questions. This way, when the training finishes, Student knows the correct answer for all of the used questions.

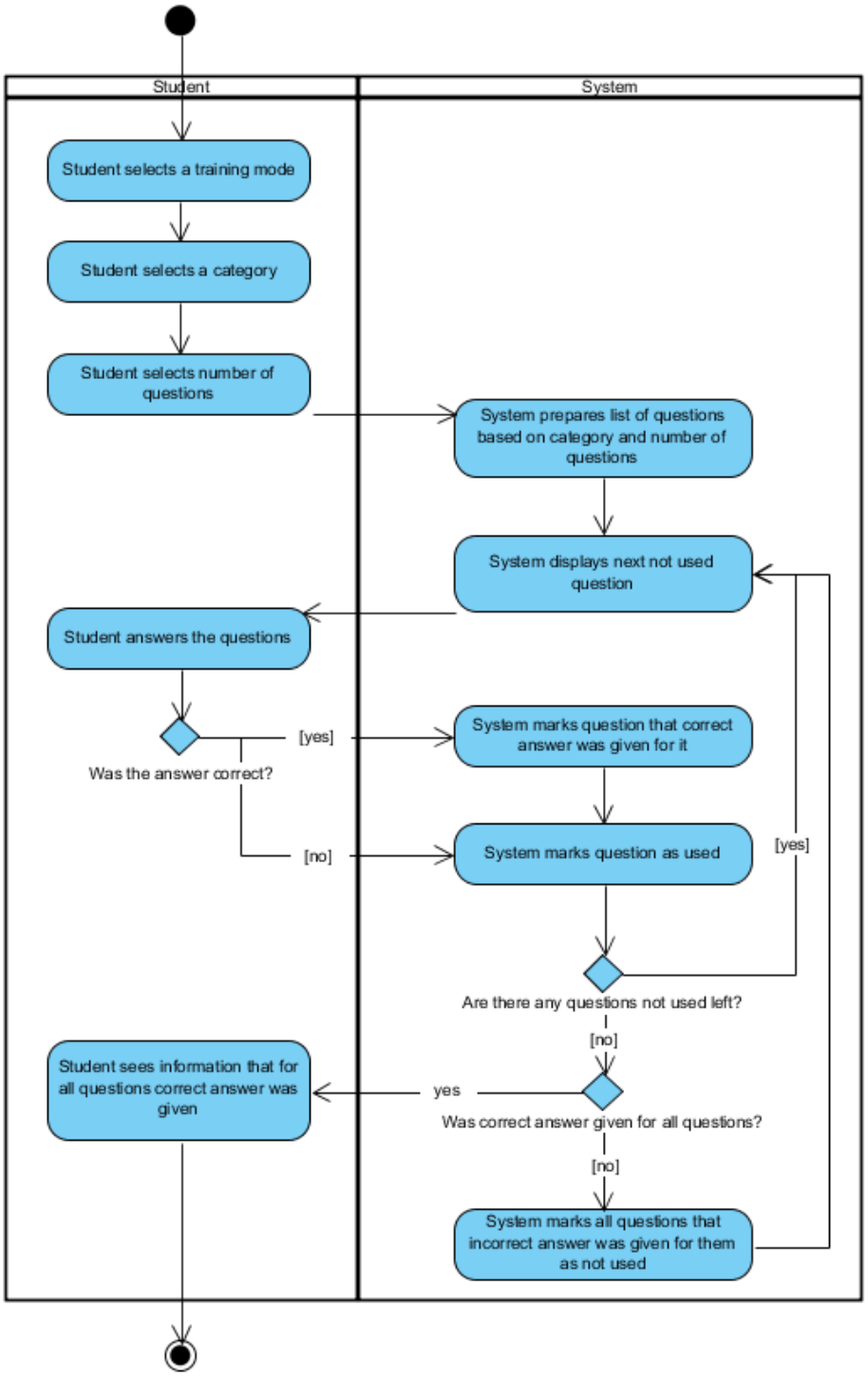

Fig. 4.9 Mechanism of questions in training mode.

4.2.2. Competition mode

Figure 4.10 illustrates mechanism of questions displaying in competition mode.

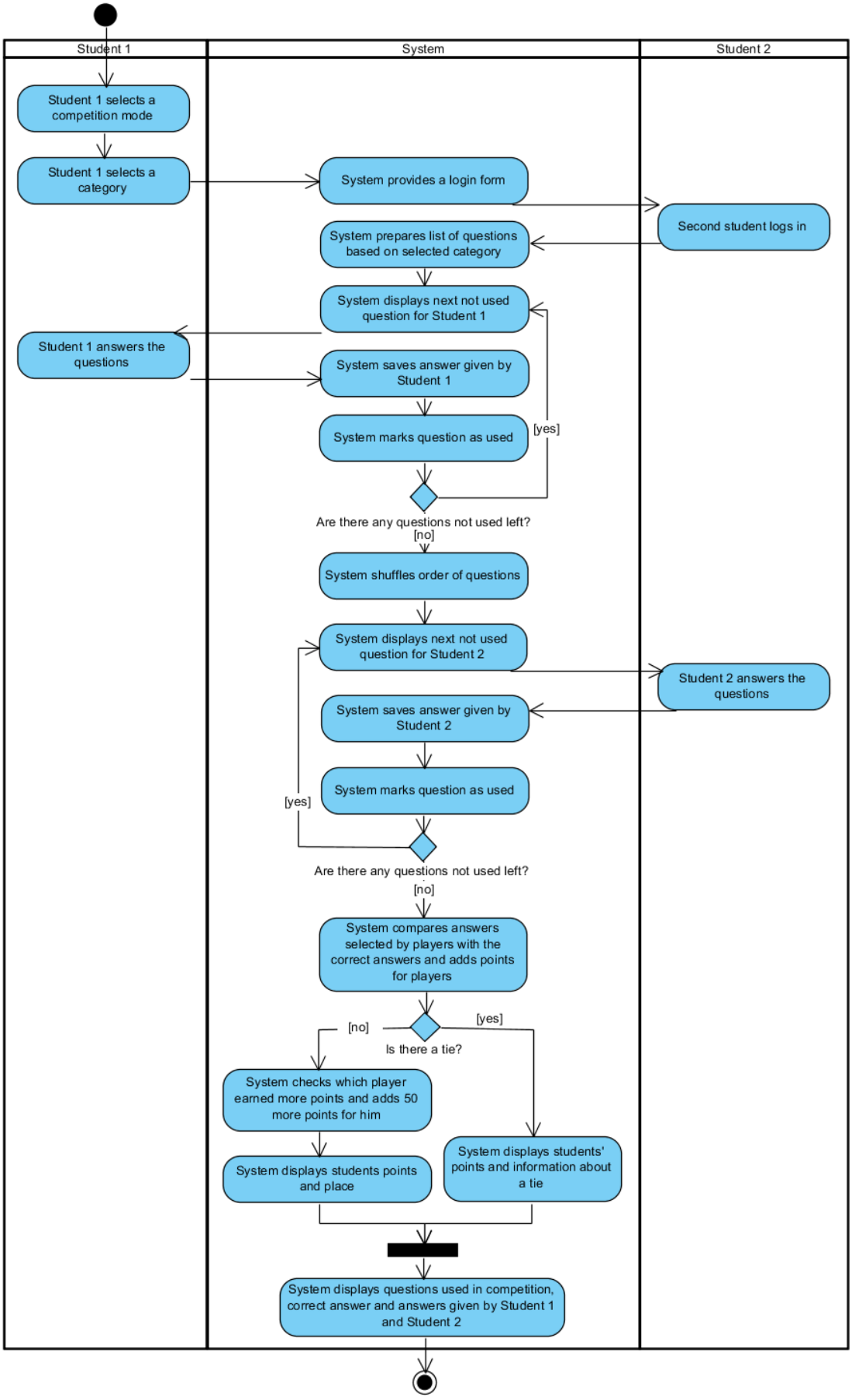

Fig. 4.10 Mechanism of questions in competition mode.

Here the most important part is that I did not wanted the same order of questions for player 1 and player 2, so I decided to shuffle the order. I did it because of the same reasons I shuffle the answers – the second player should think about what answer is correct and not just select the same one as first player did. Figure 4.10 does not include the situation when the Student selects a bonus or when the time intended for the competition runs out. In this second case, system redirects to "System shuffles order of questions" action for first player, and to "System compares answers selected by players with correct answers and adds points for them" action for second player.

4.3. Quality assurance

4.3.1. Manual tests

To test my application I used three test cases, one for each role existing in my application. I created them in form of end-to-end tests, based on use cases. Their role is to make sure that user is able to go through all features of the system successfully. Of course, during implementation I have also checked negative scenarios, like trying to use half-half bonus with Student who does not have any of them left. However, those so called "happy paths" I find the most crucial for the proper work of the application and I have used them any time I have made any major changes in the system.

I have also used my application in a real-life situation, at the extracurricular activities in the primary school with children from 6^{th} class and three teachers. This situation allowed me to check if the system works correctly when it is used by many users at the same time. Also, I have received a feedback from them. Teachers said that they think the application would be very useful for them, pointing especially that they appreciate that in training mode questions with wrong answer are returning to the Student and that answers every time have different order. Students said that they would like to use this application while preparing for the exam. Some of them said that they would change the user interface and one student also pointed out that he would like to have some explanations of the questions included, like showing how to solve an equation or definitions of the terms.

4.3.2. Tests automation

Other way to test an application is with the use of tests automation. Selenium WebDriver is one of the options. It is an Object Oriented API, which allows to make calls to the browsers, with the use of their native support for automation [17]. I used it to test the forms I have in my application (like login form or adding new question form) to check validation rules and if the correct text is displayed.

Next tool that I have used is Applitools Eyes [1]. Its purpose is to make sure that layout of the page is not broken. It supports many automation environments, including Selenium. By adding some code to test, we make screenshots of desired pages. Those screenshots are then compared with the baseline, created with the first test run. The baseline can be later changed, as well as ignore and floating regions can be specified.

Test automation tools are very useful, as they help in regression testing, to make sure that after applying changes in the application, previously implemented elements are still working correctly.

5. Instruction manual of the application

5.1. Instruction manual for Student

In order to access main features of the application, user needs to be registered into the system. Student can do it by himself, of System Administrator can create an account for him. If he decides to do it personally, from the index page he needs to choose “Zarejestruj się” button. This way register page will be opened (Fig. 5.1).

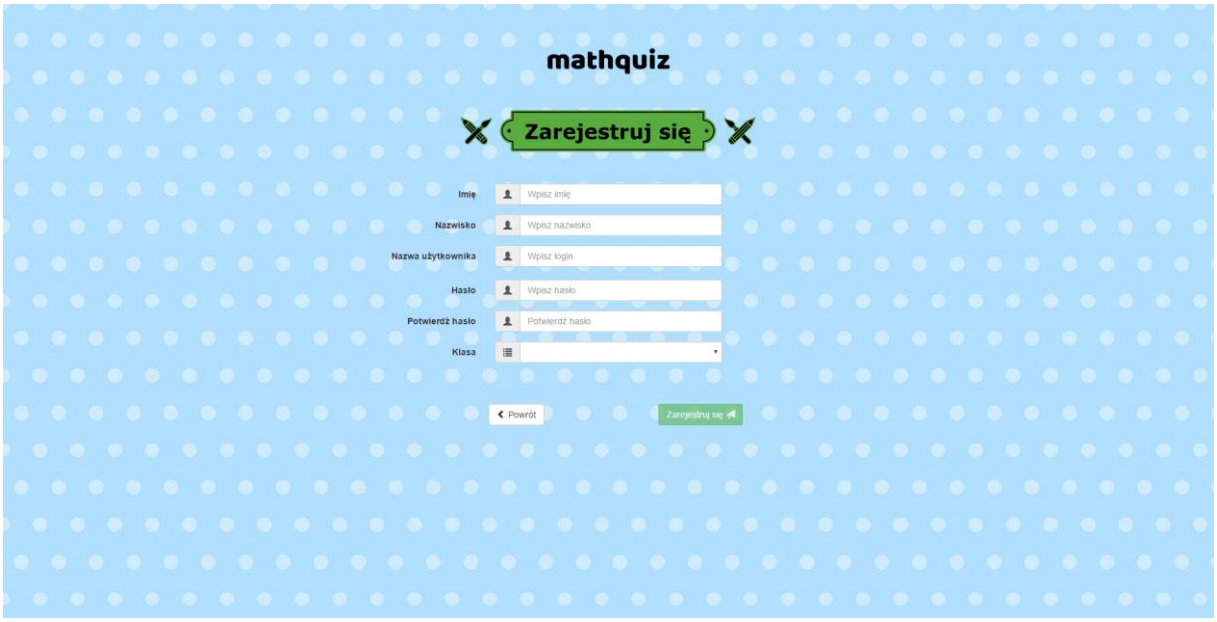

Fig. 5.1 mathquiz – register page.

To register, Student has to enter his first name, last name, login and password, confirm the password and select class. First and last name must consist of at least 2 characters, while login and password of 6 characters. When all data is entered successfully and “Zarejestruj się” button below the form is clicked, main page is displayed (Fig 5.3).

If Student is already registered into the system, on the index page he chooses “Zaloguj się” button. Login page is then displayed (Fig. 5.2), Student provides his login and password and clicks on “Zaloguj się” button below the form. Then, main page is displayed (Fig 5.3).

Fig. 5.2 mathquiz – login page.

On the main page three sections can be distinguished: navbar, carousel and mode selection. Navbar is the same in all Student's pages. Application logo in a left corner is a reference to the main page. On the right side Student can see his level, points and points needed to enter next level. Also, buttons to see statistics and to logout are placed there. Clicking on the logout button causes redirection to the login page. The carousel contains three slides: welcoming text, explanation of training mode and explanation of competition mode. Last section allows to choose mode to play.

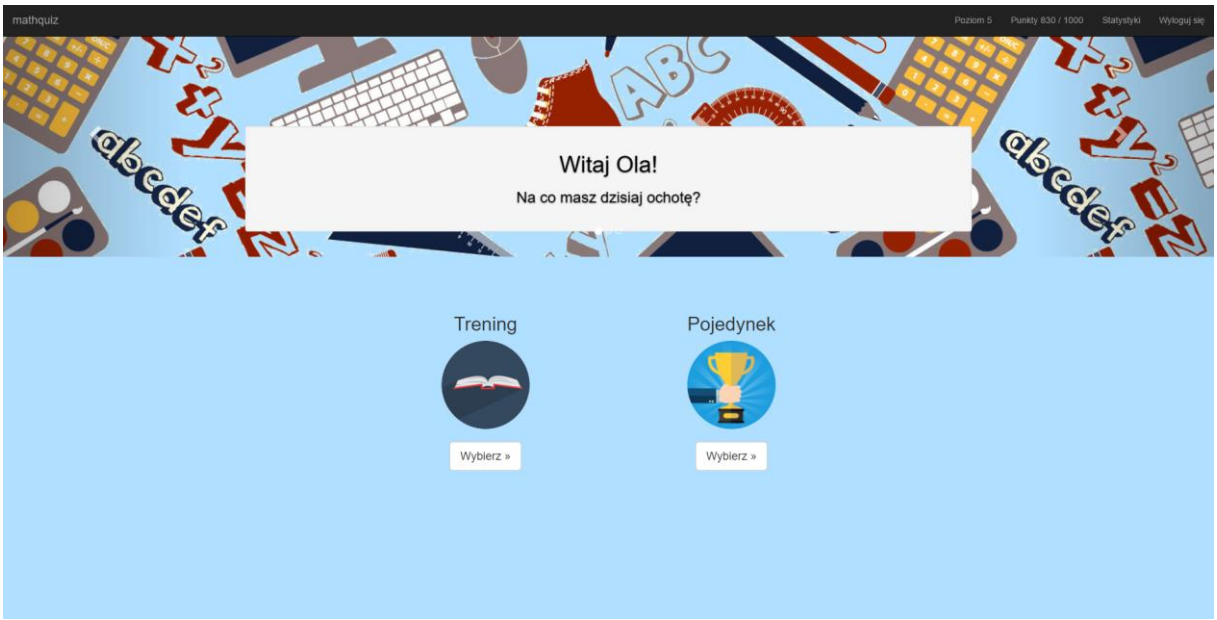

Fig. 5.3 mathquiz – Student's main page.

5.1.1. Training mode

When the Student clicks on “Wybierz” button under the “Trening” label and image, training mode is selected. Then, Student has to choose category. Figure 5.4 presents case where logged in Student is on 5th level, and has mouse over last category image. In this scenario first two categories are available and last image displays level needed to unlock final category.

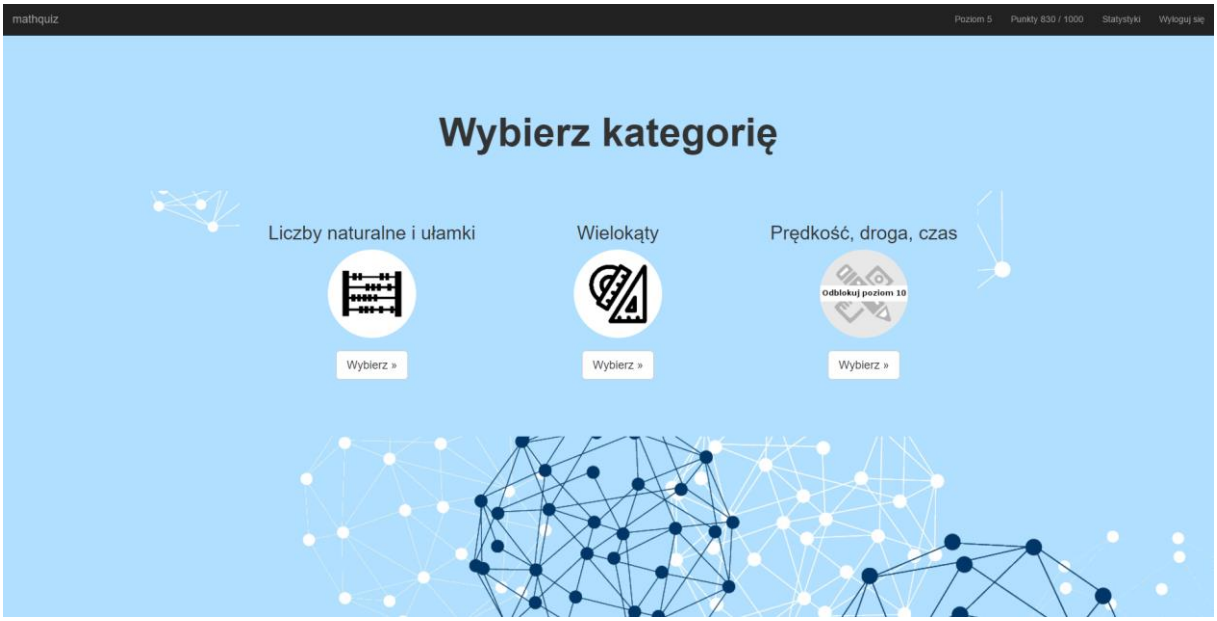

Fig. 5.4 mathquiz – choose category page.

After choosing the category, student must enter how many questions he want to practice (Fig. 5.5). Minimum number is 1, and maximum number depends on number of all questions present for selected category. Range from which numbers are accepted is displayed in the text field.

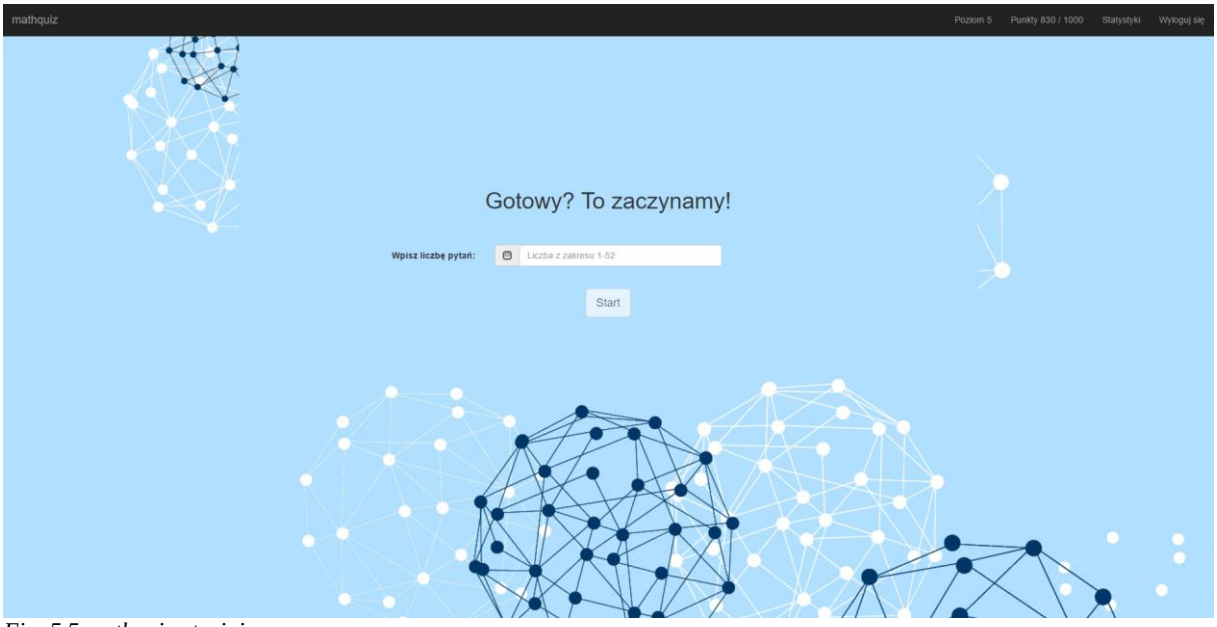

Fig. 5.5 mathquiz –training: prepare page.

When number of questions is entered, Student clicks on “Start” button and the training begins (Fig. 5.6). Student sees question and four answers. Above, there is a progress bar which indicates how close the Student is to finish the training. To finish the training, all questions must have correct answer selected. After selecting one of the answers, Student clicks on “Następne pytanie” button. If the selected answer was correct, progress bar will move, and if not, it will not change. When only one questions is left, “Następne pytanie” button changes to “Koniec” button. If Student selects correct answer and clicks on this button, summary page is displayed (Fig. 5.7).

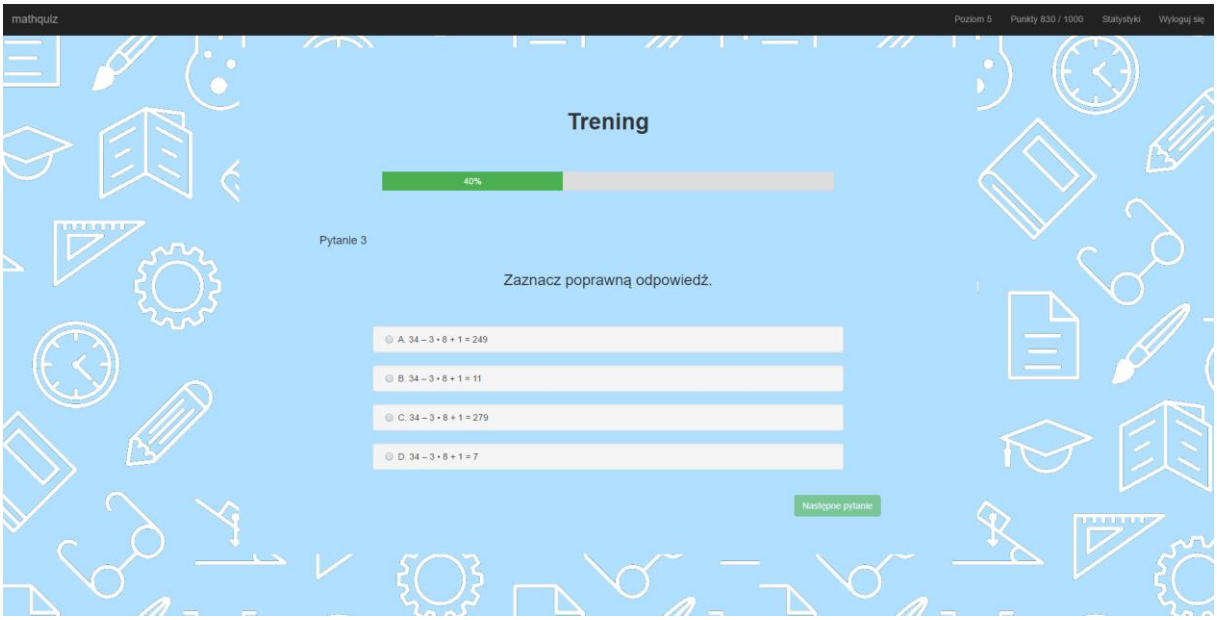

Fig. 5.6 mathquiz – training page.

Fig. 5.7 mathquiz – training: summary page.

5.1.2. Competition mode

If the Student selects “Wybierz” button under the “Pojedynek” label and image, he starts competition mode. First, he needs to select category, which is the same process as in the case of training mode. Then, second player has to enter his data (Fig. 5.8). System validates if the data is correct and if role of the second player is Student.

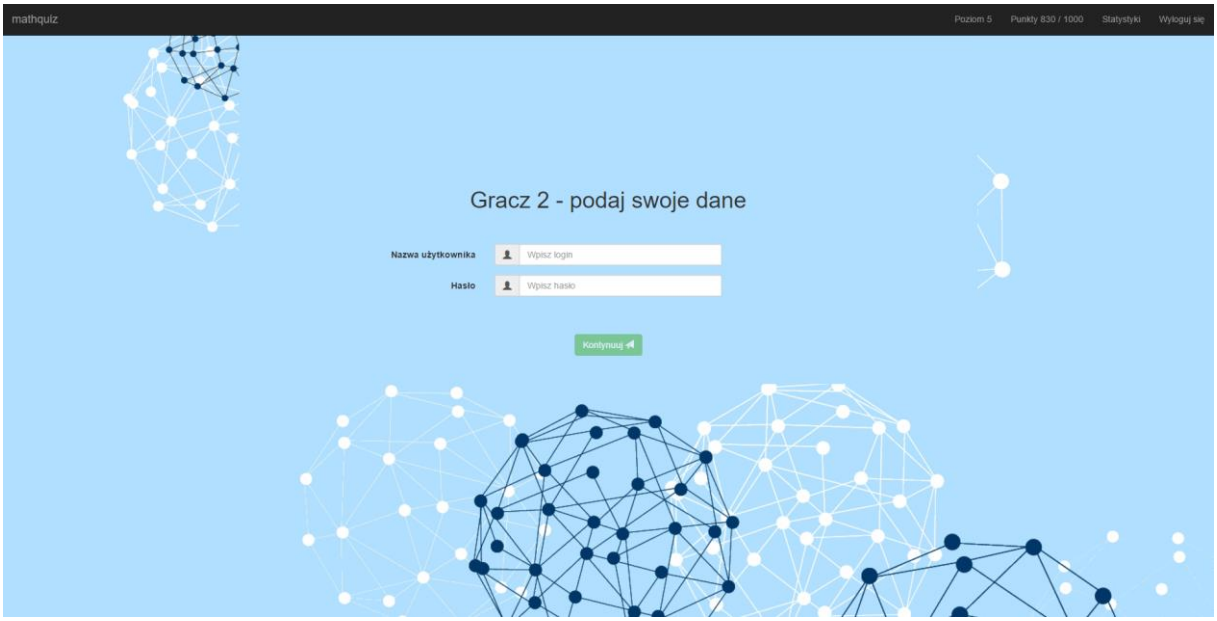

Fig. 5.8 mathquiz – competition: second player login page.

After entering data, Student clicks on “Kontynuuj” button. When first player is ready, he clicks on “Start” and competition begins (Fig 5.9). Similarly to the training, Student sees question and four answers, however here the progress bar shows how many time has passed since the beginning of competition, for each player separately. If the progress bar is fully filled, redirection is performed – in case of first player to the competition for the second player and in case of the second player to the summary page (Fig.5.11).

Below the progress bar, on the right side are bonuses. Student sees how many of them he has, and when hovering with mouse over them, sees tooltips explaining their action.

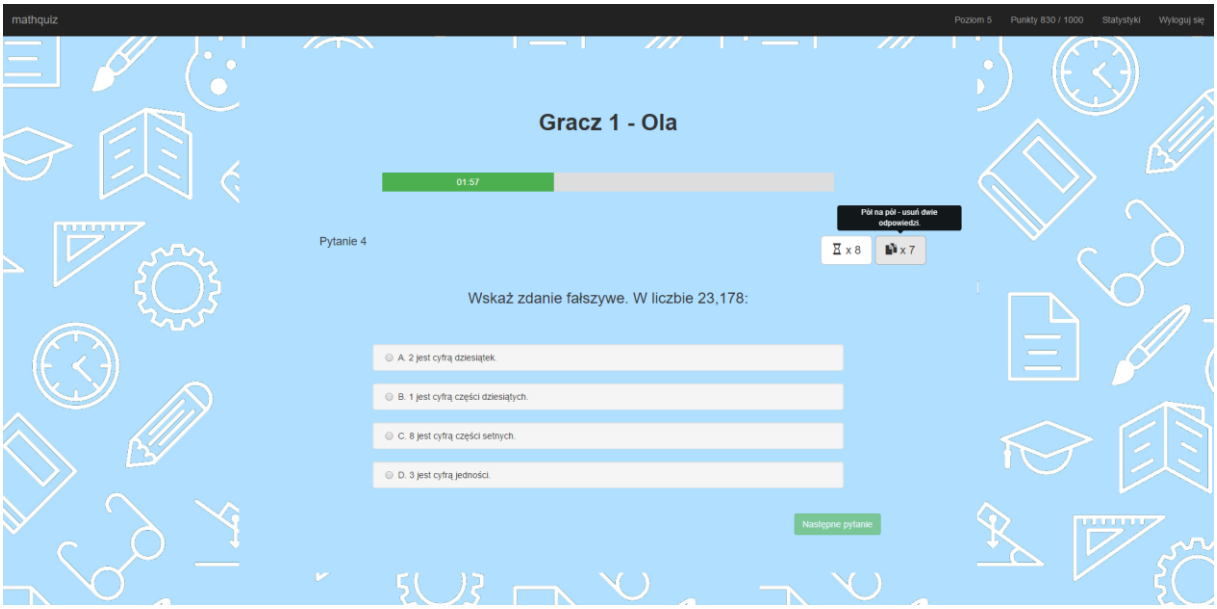

Fig. 5.9 mathquiz – competition.

If the Student decides to use half-half bonus, two answers disappear and two – one correct and one incorrect are left (Fig. 5.10).

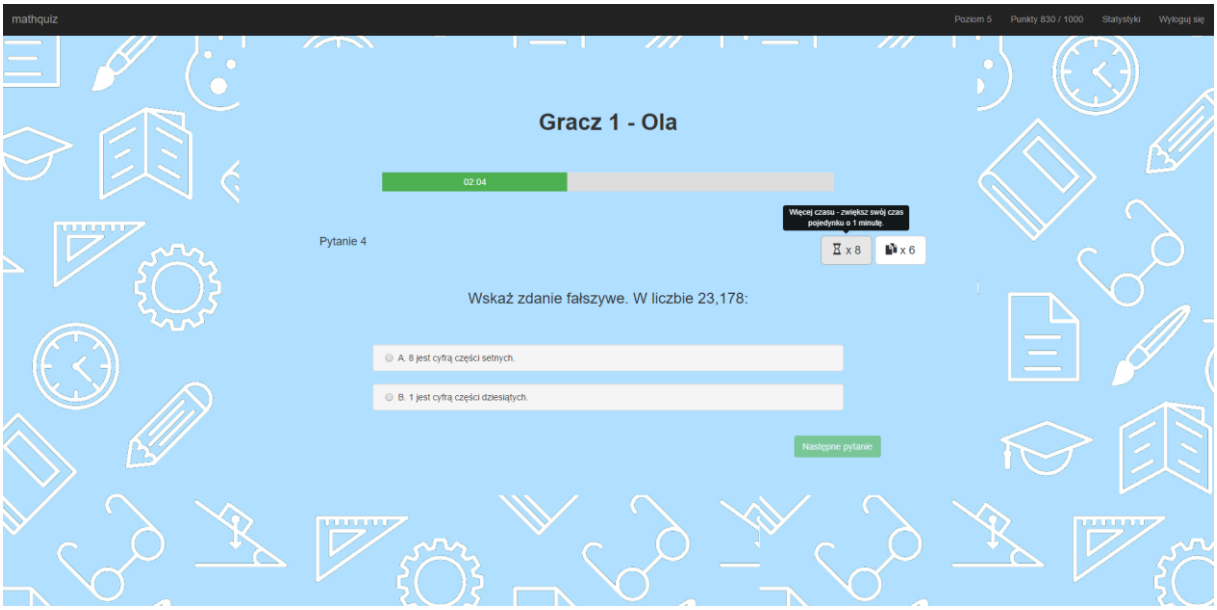

Fig. 5.10 mathquiz – competition: half-half bonus.

When both students finished the competition, summary page is displayed (Fig. 5.11). Figure 5.11 presents the case when one player earned more points than the second one. However, it is also possible that the competition finished with a tie.

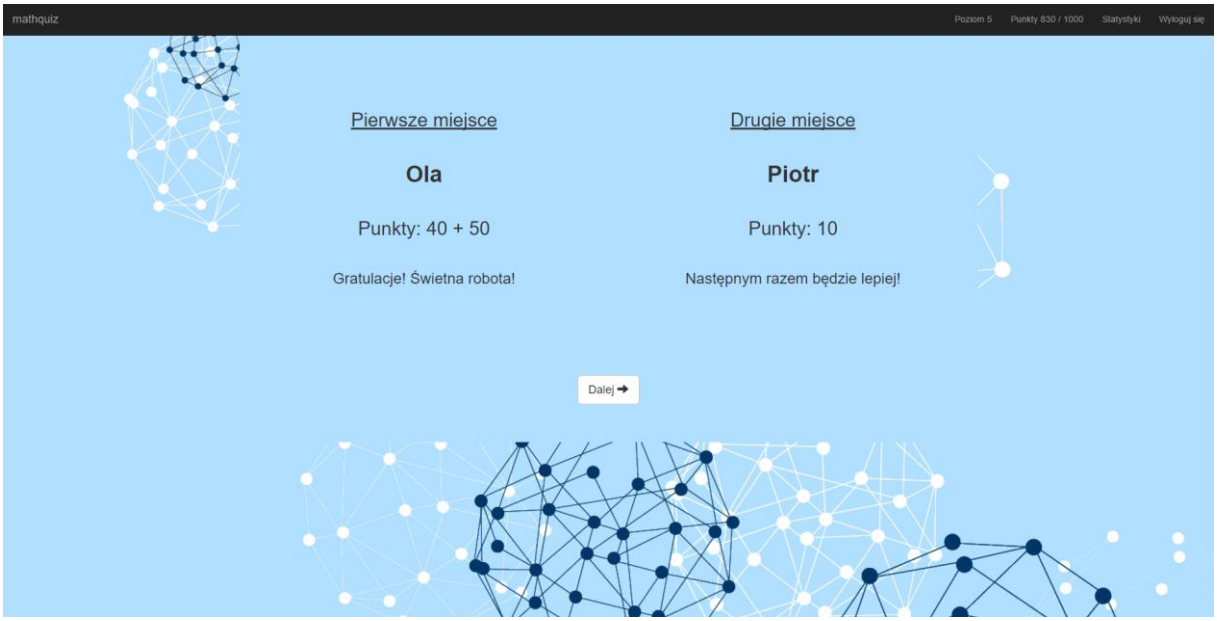

Fig. 5.11 mathquiz – competition: summary page.

After clicking on the "Dalej" button, table with questions used in competition is displayed, together with correct answers and the answers selected by players (Fig. 5.12).

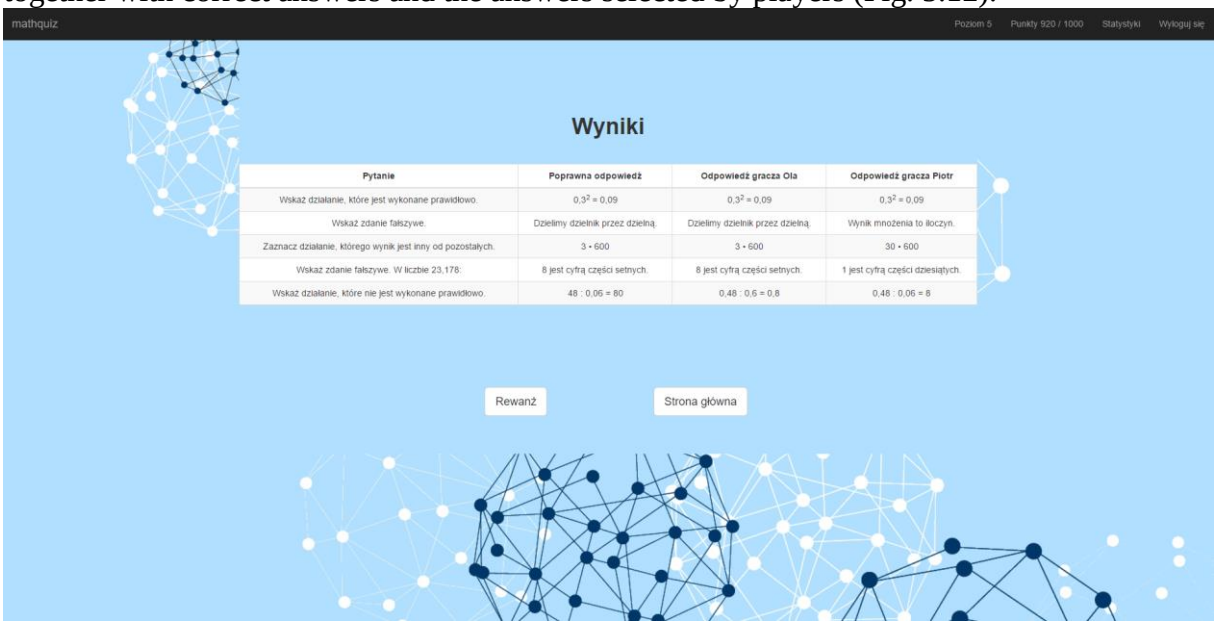

Fig. 5.12 mathquiz – competition: results page.

5.1.3. Statistics

From the navbar Student has the possibility to enter the statistics page (Fig. 5.13). There he can see all the pupils from his class that are registered into the system, together with their level and points.

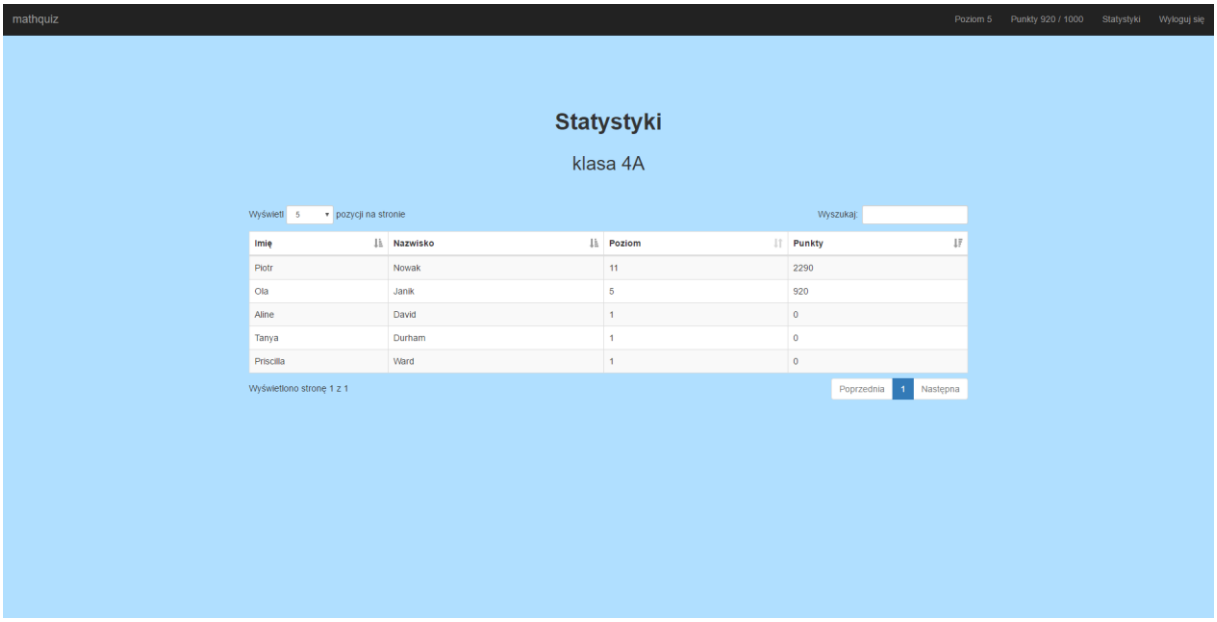

Fig. 5.13 mathquiz – statistics for Student.

5.2. Instruction manual for Teacher

Teacher logs in to the application the same way as Student does. After doing that, his main page is displayed. Similarly as for Student, this page is divided into three areas, however instead of mode selection, he can choose from manage questions and view statistics actions. Those functionalities can be also accessed from the navbar.

5.2.1. Manage questions

Clicking on the “Wybierz” button under the “Zarządzaj pytaniami” label and image or choosing “Pytania” from the navbar opens view questions page (Fig. 5.14). By default questions from all categories are displayed, however above the table on the left side, Teacher can choose to display only questions form specified category. On the right side, in the text field next to “Wyszukaj” label, user can enter desired text and search the table. Clicking on any of the column’s headers will cause sorting by this column.

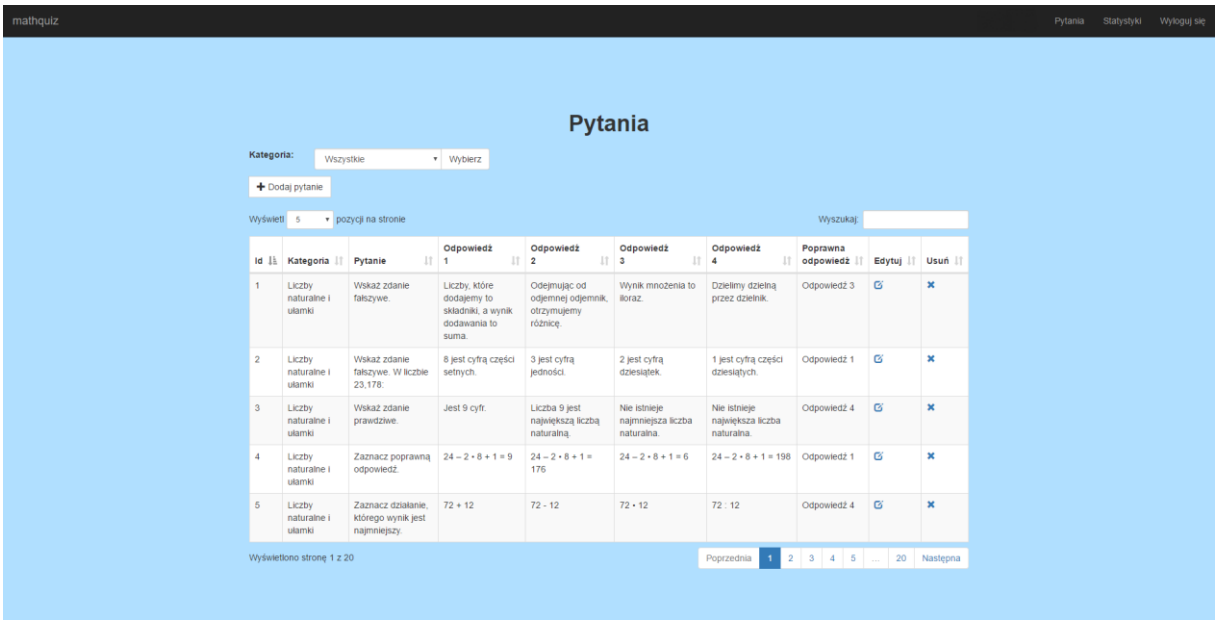

Fig. 5.144 mathquiz – view questions page.

Above the table on the right side there is “Dodaj pytanie” button. It opens form for adding new question (Fig. 5.15). Teacher has to choose new question’s category, enter question, four answers and correct answer. For all text fields’ validation is set - at least 2 characters must be entered.

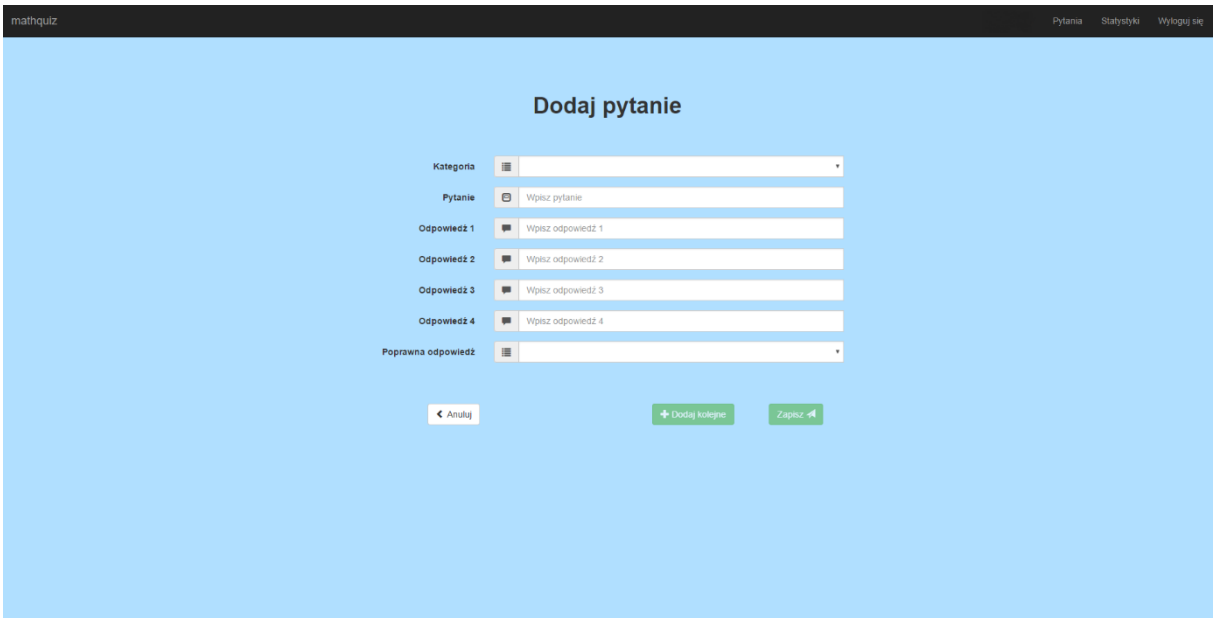

Fig. 5.15 mathquiz – add question page.

In each row, in the “Edytuj” column there is displayed icon. Clicking on it causes to display form for editing the question (Fig. 5.16). By default, all fields are filled with question’s current data. Validation rules also apply here. Clicking on the icon in “Usuń” column and confirming in the popup window causes deleting of the question.

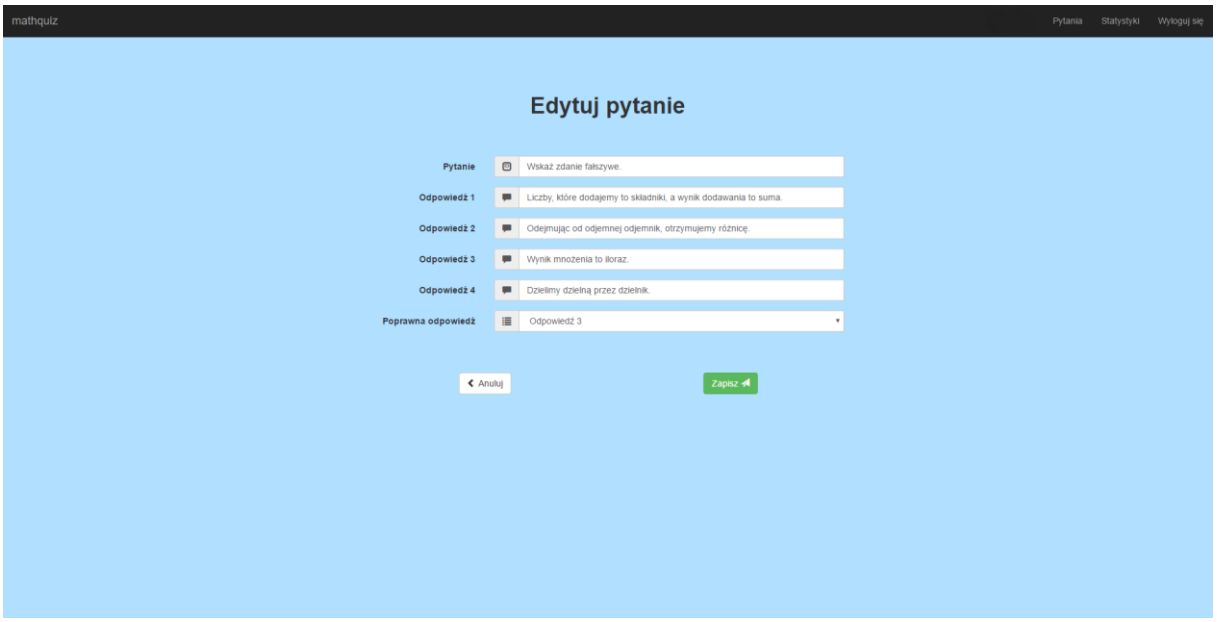

Fig. 5.16 mathquiz – edit question page.

5.2.2. Statistics

Teacher can also see all Students' statistics (Fig. 5.17). Sorting and searching works as previously described. Filtering can be done by Students' class.

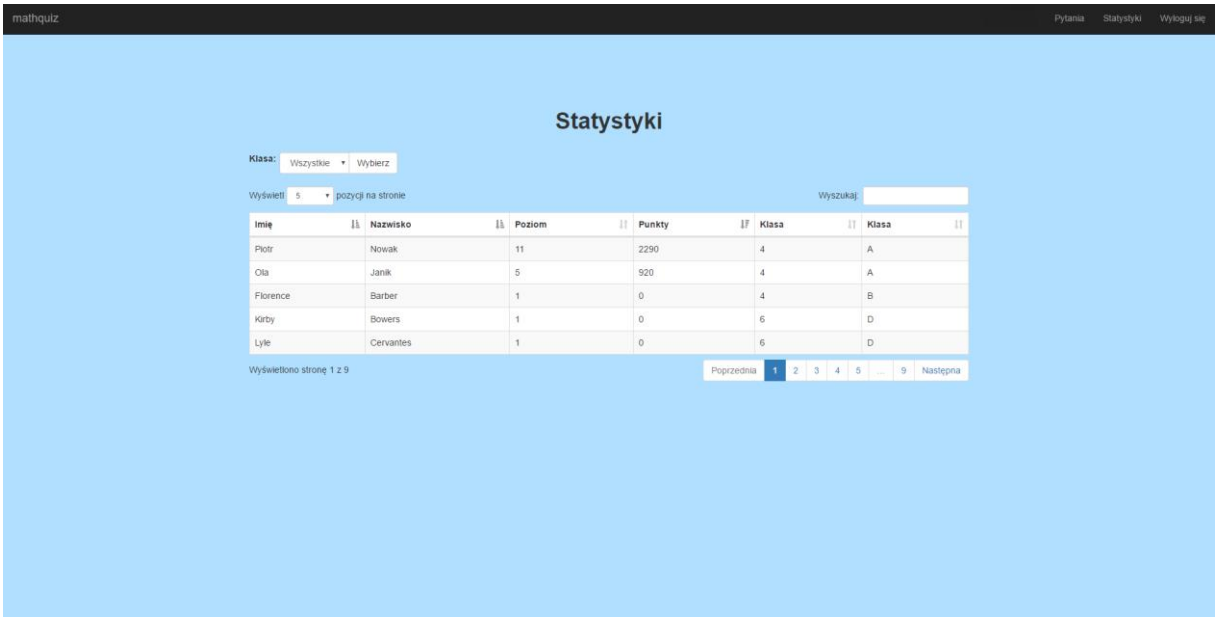

Fig. 5.17 mathquiz – statistics for Teacher.

5.3. Instruction manual for System Administrator

System Administrator has all the functionalities that Teacher has extended with the possibility to manage users. After logging to the system Administrator sees his main page (Fig. 5.18). It is similar to Teacher's one except that "Zarządzaj użytkownikami" functionality is added in the row where "Zarządzaj pytaniami" and "Zobacz statystyki" are as well as in the navbar.

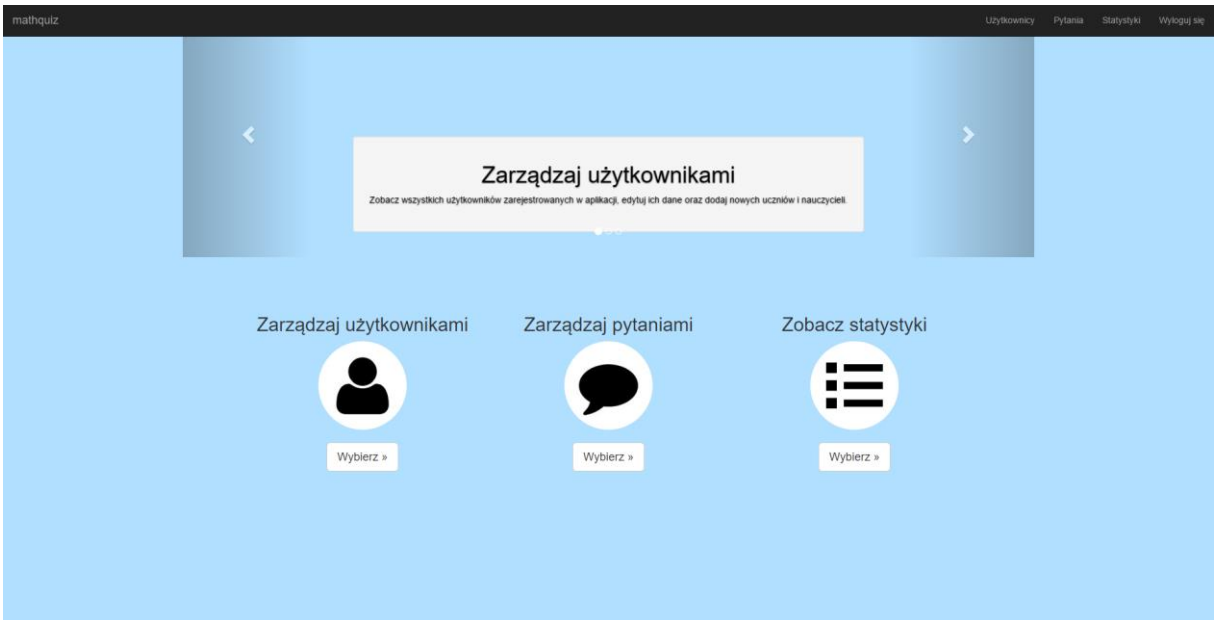

Fig. 5.18 mathquiz – System Administrator's main page.

5.3.1. Manage users

If the System Administrator clicks on "Wybierz" button under the "Zarządzaj użytkownikami" label and image or "Użytkownicy" button from the navbar, list of all users registered into the system is displayed (Fig. 5.19). Searching and sorting works the same way as in view questions list, but here filtering can be done by role of user.

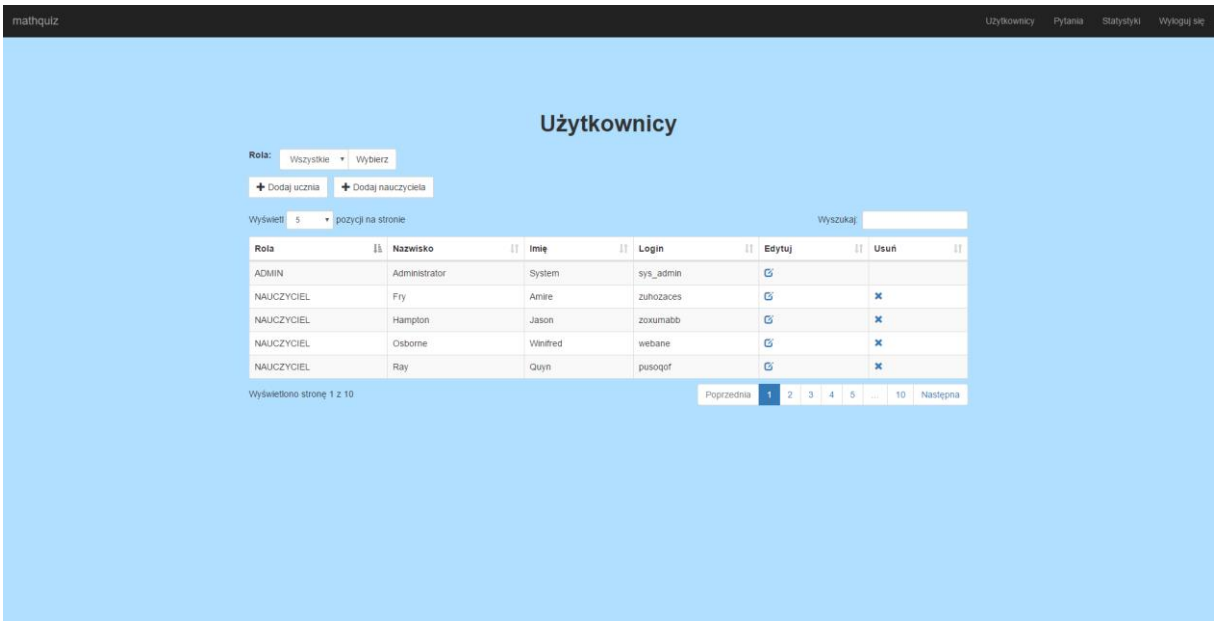

Fig. 5.19 mathquiz – view users page.

Clicking on the "Dodaj ucznia" and "Dodaj nauczyciela" buttons will cause displaying proper form for adding new user (Fig. 5.20). Validation rules are the same as on the registration page.

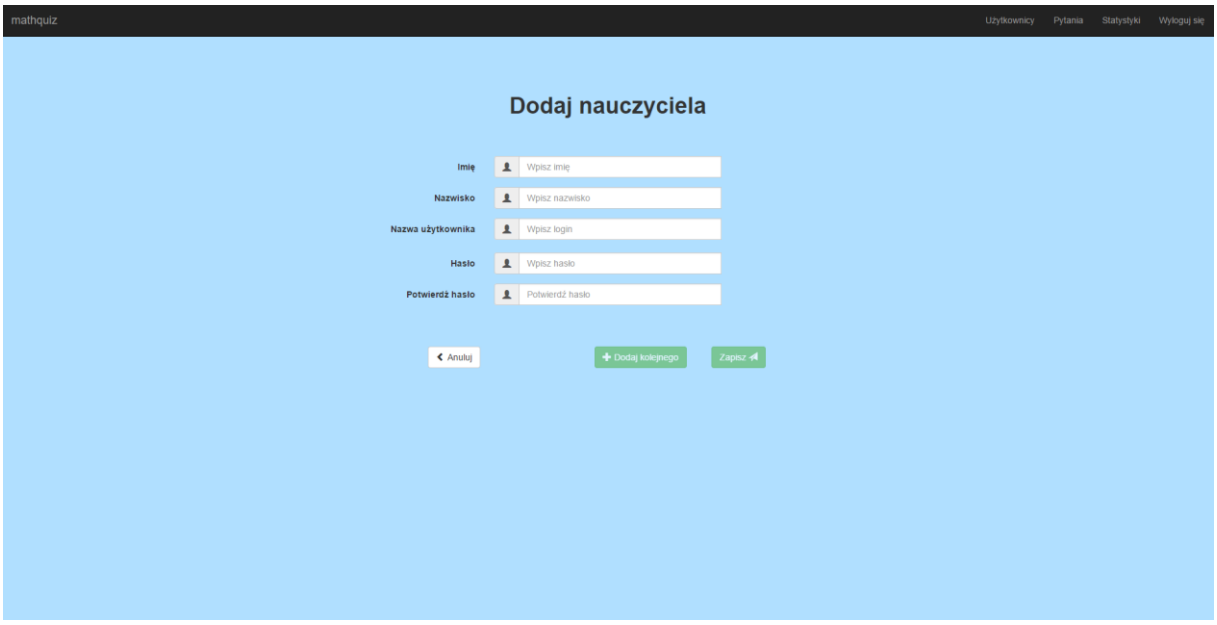

Fig. 5.20 mathquiz – add Teacher page.

Editing (Fig. 5.21) and deleting a user works the same way as in case of questions. System Administrator cannot be deleted.

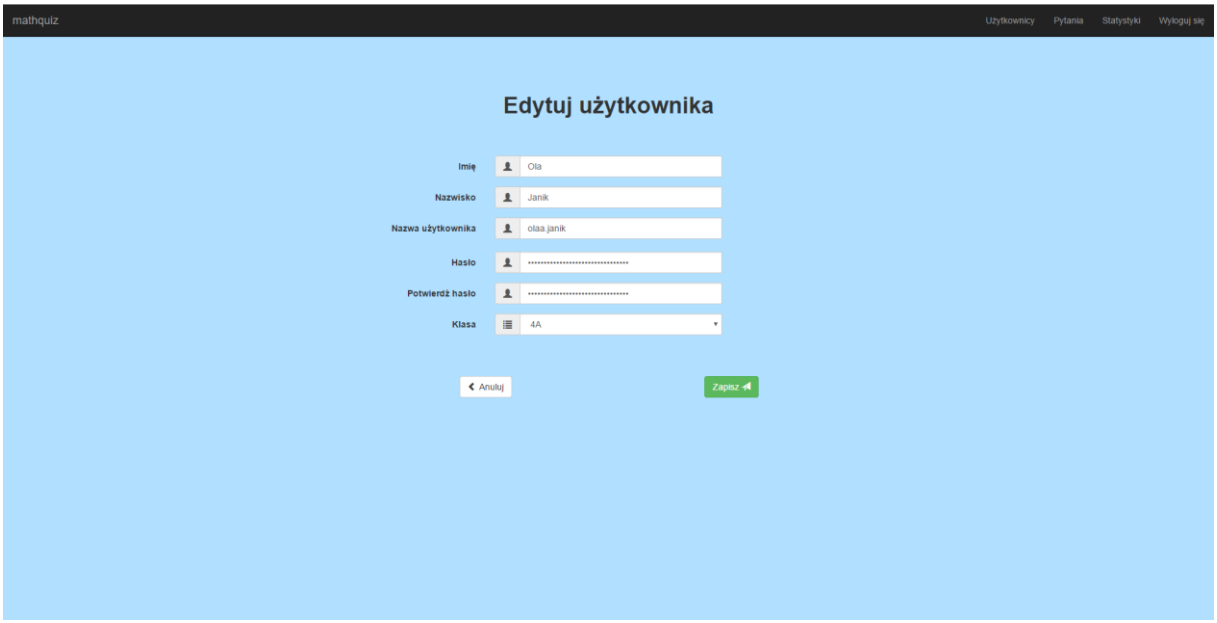

Fig. 5.21 mathquiz – edit Student page.

6. Summary

6.1. Conclusions

The goal of the work was to design and implement web-based educational game with gamification supporting the work of the teacher, with RWD and flat design. Realization of the work consisted of creating vision of the application, formulating requirements and implementing working system. Gamification mechanisms were described and some of them were implemented. With the use of Bootstrap responsiveness of the application was provided.

Created application allows teachers to enrich common lessons with a new tool. It is an interesting alternative to written tests as well as it saves teachers a lot of time by removing the need to check the answers manually. Because of using gamification techniques the system is more appealing to the students. It will not cause that children will start love mathematics, however I hope it will help at least some of them to learn this subject.

There are many areas were the application can be ameliorated and extended. However the requirements of the work were fulfilled. Its product is working system called mathquiz.

6.2. Future work

As mentioned, the created system can be improved. To make the use of the application more pleasant, following features I would like to add in the future:

- Implement badges, which Students can obtain by answering correctly on 5 questions in a row in training mode, by playing every day for a week, etc.
- Add popup window when Student enters next level, with the possibility to choose one of three chests which will contain random bonuses.
- Add immediate feedback for Student playing in training mode, to let him know if he selected correct or incorrect answer.
- Make Student position on the statistics page highlighted.
- Make it possible for teacher to add new category.
- Make competition time and number of questions used in competition customizable by a teacher, for each class separately.
- Add more types of questions, like drag and drop, complete the sentence, etc.
- Add theory for the categories, so that students can remind themselves the material before taking part in training or in competition.

Bibliography

[1] Applitools
https://applitools.com/, Retrieved 2017-05-27

[2] Beaird J., *Niezawodne zasady Web designu.* Helion S.A., 2012.

[3] BootstrapValidator
http://bootstrapvalidator.votintsev.ru/, Retrieved 2017-05-21

[4] Chaffey D., *Mobile Marketing Statistics compilation*, Smart Insights
http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/, Retrieved 2017-05-14

[5] DataTables
https://datatables.net/, Retrieved 2017-05-21

[6] Duolingo
https://www.duolingo.com/, Retrieved 2017-05-07

[7] *Dzieci w sieci*, Gemius Polska
http://www.gemius.pl/wszystkie-artykuly-aktualnosci/dzieci-w-sieci.html, Retrieved 2017-05-06

[8] Flaticon
http://www.flaticon.com/, Retrieved 2017-05-14

[9] Freepik
http://pl.freepik.com/, Retrieved 2017-05-14

[10] Kahoot!
https://getkahoot.com/, Retrieved 2017-05-07

[11] Khan Academy
https://pl.khanacademy.org/, Retrieved 2017-05-07

[12] LearningApps.org
https://learningapps.org/, Retrieved 2017-05-07

[13] Lovato N., *16 Reasons Why Players Are Leaving Your Game*, GameAnalytics
http://www.gameanalytics.com/blog/16-reasons-players-leaving-game.html, Retrieved 2017-05-06

[14] May T., Clum L., *The beginner's guide to flat design*, Creative Bloq
http://www.creativebloq.com/graphic-design/what-flat-design-3132112, Retrieved 2017-05-14

[15] Peters J., *5 Main Advantages of Game-Based Learning*, Bright Hub Education
http://www.brighthubeducation.com/teaching-methods-tips/129304-advantages-of-game-based-learning/, Retrieved 2017-05-06

[16] PHP 5 Tutorial, w3schools.com
https://www.w3schools.com/php/default.asp, Retrieved 2017-05-21

[17] *Selenium Documentation - Selenium WebDriver*, SeleniumHQ
http://www.seleniumhq.org/docs/03_webdriver.jsp, Retrieved 2017-05-22

[18] Squla
https://www.squla.pl/, Retrieved 2017-05-07

[19] Zichermann G., Cunningham C., *Gamification by Design. Implementing Game Mechanics in Web and Mobile Apps.* O'Reilly Media, Inc., 2011.

List of figures

Fig. 2.1 Bartle's player types (source: [19]).... 10
Fig. 2.2 Level complexity (source: [19]).... 11
Fig. 2.3 Social engagement loop (source: [19]). 12
Fig. 2.4 Learning Apps - Examples of applications in category Mathematic (source: [12]). .. 13
Fig. 2.5 Learning Apps - Example of created application (source: [12]).... 14
Fig. 2.6 Khan Academy - Exercise after video tutorial (source: [11]).... 15
Fig. 2.7 Khan Academy - Popup after finish of exercise in mission (source: [11]).... 15
Fig. 2.8 Kahoot! - Preview of created quiz before start (source: [10]). 16
Fig. 2.9 Kahoot! - Preview of created quiz during the game (source: [10]). 17
Fig. 2.10 Duolingo - Ladder modules (source: [6]). 18
Fig. 2.11 Duolingo - Shop (source: [6]).... 18
Fig. 2.12 Squla – Different categories (source: [18]).... 19
Fig. 2.13 Squla – Shop (source: [18]). 20
Fig. 3.1 Multi-platform device use 2017 (source: [4]).... 23
Fig. 3.2 Use-case diagram.... 24
Fig. 3.3 Database schema.... 36
Fig. 4.1 Index page on desktop computer. 38
Fig. 4.2 Index page on mobile phone.... 39
Fig. 4.3 Training on the desktop computer. 39
Fig. 4.4 Training on mobile phone – real view. 40
Fig. 4.5 Training on mobile phone – full view.... 40
Fig. 4.6 Expanded navbar on mobile phone.... 40
Fig. 4.7 Fragment of registration page with validation errors.... 41
Fig. 4.8 View list of users page.... 42
Fig. 4.9 Mechanism of questions in training mode.... 43
Fig. 4.10 Mechanism of questions in competition mode. 44
Fig. 5.1 mathquiz – register page.... 47
Fig. 5.2 mathquiz – login page.... 48
Fig. 5.3 mathquiz – Student's main page.... 48
Fig. 5.4 mathquiz – choose category page. 49
Fig. 5.5 mathquiz –training: prepare page.... 49
Fig. 5.6 mathquiz – training page.... 50
Fig. 5.7 mathquiz – training: summary page.... 50
Fig. 5.8 mathquiz – competition: second player login page.... 51
Fig. 5.9 mathquiz – competition.... 52
Fig. 5.10 mathquiz – competition: half-half bonus. 52
Fig. 5.11 mathquiz – competition: summary page. 53
Fig. 5.12 mathquiz – competition: results page. 53
Fig. 5.13 mathquiz – statistics for Student.... 54
Fig. 5.144 mathquiz – view questions page. 55
Fig. 5.15 mathquiz – add question page.... 55
Fig. 5.16 mathquiz – edit question page.... 56
Fig. 5.17 mathquiz – statistics for Teacher. 56
Fig. 5.18 mathquiz – System Administrator's main page. 57
Fig. 5.19 mathquiz – view users page. 57
Fig. 5.20 mathquiz – add Teacher page. 58
Fig. 5.21 mathquiz – edit Student page.... 58

List of tables

Table 2.1 User Summary 9
Table 3.1 UC01: View users 25
Table 3.2 UC02: Add Student 26
Table 3.3 UC03: Add Teacher 26
Table 3.4 UC04: Edit User 27
Table 3.5 UC05: Delete Student 28
Table 3.6 UC06: Delete Teacher 28
Table 3.7 UC07: View questions 29
Table 3.8 UC08: Add question 29
Table 3.9 UC09: Edit question 30
Table 3.10 UC10: Delete question 31
Table 3.11 UC11: Take part in training 31
Table 3.12 UC12: Take part in competition 32
Table 3.13 UC13: Choose number of questions 33
Table 3.14 UC14: Choose category 33
Table 3.15 UC15: View statistics 34
Table 3.16 UC16: Log in 35
Table 3.17 UC17: Register 35

Attachments

[1] CD with copy of thesis in PDF format and source code of application.