

Design and development of an Android-based quiz application

Elena Bautu, Cristina Draghici

Ovidius University Constanta
124 Mamaia Blvd., Constanta, Romania
E-mail: ebautu@univ-ovidius.ro

Abstract. The increasing interest towards mobile technologies makes way for a lot of potential for software applications to solve more or less trivial problems, literally by just one “touch”. This paper focuses on taking this technological advantage and using it in the educational sector. We present the design and development of Quizriffic – a mobile application dedicated to students and professors to help them better prepare for classes. With the help of this application, professors can create quizzes test their students' knowledge. Students can practice quizzes or compete in the simulated environment of a quiz. Their progress is monitored accordingly and displayed in the form of several types of charts, so that the students can be motivated to improve their overall grades, and get a glimpse of how their efforts turn into real better results. The application also provides a common ground where professors and students can virtually meet and discuss subjects or ideas, making it easier for them to connect and keep in touch.

Keywords: Quiz-taking application; Learning management system; Game-based learning; Mobile educational platform.

1. Introduction

Nowadays, people spend a significant amount of time on their mobile phones, as there are a variety of apps that can fulfill their needs. From basic activities people do in their leisure time (watching movies, playing games, talking to their friends) to more serious ones like paying bills, translating documents or making presentations, mobile apps are there to make these activities easier and improve users' life. The fast-paced world children grow up in, having a school-related mobile application can be the key to creating a strong bonding between students and school.

Quizriffic is a solution we propose in order to improve the way students prepare for a quiz. It eases the evaluation procedure for the professors and provides easy to grasp metrics of the learning progress, useful statistics on

the educational process and, overall, increased satisfaction of both students and teachers. This paper is an extended version of the paper presented at RoCHI 2019 (Bautu, E. and Draghici, C., 2019).

1.1. Types of quizzes

From the beginning of education, a clear structuring format has proven to be a good approach to refine knowledge and abilities: studying and assessment. At the moment, there are multiple forms of assessing one's capabilities, many of which can be ultimately coordinated by a machine.

Multiple-choice questions are a popular format among many international standardized examinations, such as the SATs (Scholastic Assessment Test) or TOEFL (Test of English as a Foreign Language) (Raymond, M.R. et. al. 2019, Krosnick, J. A. 2018). A question of this type requires a set of possible answers shown to the examinee, among which one or more are correct and the others serve as distracters. A common rising problem could be the lack of testing whether the student did some reasoning while coming up with the answer or he randomly picked it and got lucky. To avoid this, a 'guessing penalty' can be introduced, so that any attempt at choosing an answer without careful consideration would be discouraged (Raymond, M.R. et. al. 2019).

Fill-in-the-blanks questions provide another interesting approach to test more accurately what the examinee is thinking. This type of question requires the learner to complete a given sentence that is missing certain words. Further hints can be attached to help at deciphering the correct answer (Krosnick, J.A. 2018). Match-the-pair and sorting/labeling quizzes can also be easily implemented in a software program to reinforce studied concepts. Students are provided with a list of certain notions and have to correctly pair them with members from a second related list, sort them in the correct order or classify them under the correct label (Krosnick, J.A. 2018). Open-ended questions seem to be the most intriguing, as completely letting a machine to examine and grade it might be a difficult task. An essay, a short paragraph, even a few sentences can be written in such manner that a pre-recorded set of answers in the database would not match, even though the base idea is similar. In our application, we use multiple-choice questions, in order to provide a certain degree of flexibility for both professors and student (see Figure 3).

2.2 Related work

In spite of the fact that there are plenty of quiz-taking related applications and games, most of them are designed merely for amusement, neglecting educational purposes. Those that truly focus on the latter are limited by the range of categories they provide.

Out of desire to promote education through this medium, a group of researchers at the Institute of Computing and Information Technology Gomal University in Pakistan have written a paper on implementing such a Project (Asghar, M.Z. et. al. 2016). Using MIT App Inventor software, they created a multiple-choice quiz app for Android that students could use to select from three categories (Computer, Verbal, Analytical) and their corresponding sub-categories (for Computer → Operating Systems, Computer Networking, etc.) to answer related questions. A question can have only one correct answer and offers one-time special features like hints, pausing or skipping. However, a pre-existing question bank is needed for this application to run smoothly.

Another similar approach was taken in the project of developing a quiz system in Android using Java and SQLite (Jiping, L., Xiao, H., Zhang, H. 2019), focused on testing knowledge of the C++ programming language. The app makes use of a question bank of multiple-choice, true-or-false and fill-in the-blanks types of questions. It features a timer and scoring system, as well as the option to review the wrong answers given by the examinee to learn from his/her mistakes. A prototype called *Quiz It* incorporates a mobile module for the lecturer, one for the student, as well as a web environment meant for administrative tasks (Adam, D. et. al, 2014).

Taking quizzes, doing assignments, studying courses anywhere, anytime is equivalent to a ubiquitous learning environment in which students can be completely immersed. There is a related project on this matter created for the Android platform that retrieves courses, tests, grades from the database through a third-party library based on the SOAP architecture (Shanmugapriya, M., and Tamilarasi, A. 2011).

In the remaining of this section, two popular mobile-based applications that promote enriching one's knowledge levels by taking quizzes in a gamely fashion are presented.

QuizUp is a trivia game developed for mobile, where users can play against one of their friends or even against a random person from around the world in real time (QuizUp, September 2019).

Serving as a game-based learning platform, *Kahoot!* is widely used in schools or other educational institutions. It contains multiple-choice quizzes that can be used to assess students' knowledge. Designed for social learning, Kahoot requires students to gather around a screen in order to view the questions along with their corresponding answer options, and select the chosen answer from their own devices. To access a certain quiz, players have to insert a PIN generated when creating the quiz lobby. The teacher is the one that creates the questions and establishes how many points each correctly answered question has. The points are then awarded to the students that chose the correct answer, and are also shown after each question to form a leader board (Kahoot, September 2019).

2. The proposed system

Our application is meant to be used by professors and students altogether. From the professor's point of view, the aim of Quizrific is to provide the freedom to create customized quizzes for students to take at a scheduled date and time, eliminating the time-consuming after-work activities like grading test papers. Another advantage is that the professor can easily track the progress of his/her students, based on current and previous quiz results. The main technologies used in the development are Java, PHP, MySQL, node.JS, JavaScript and socket.io. Figure 1 depicts the architecture of the system.

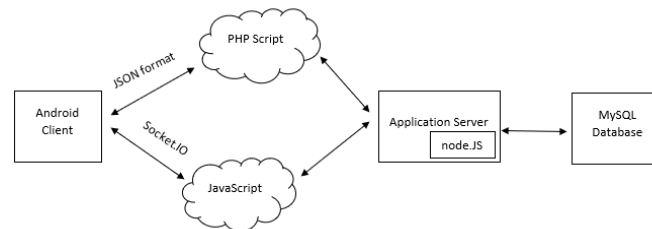


Figure 1. The architecture of the proposed system.

A *software actor* is a role played by an entity that interacts with the application. The actors for Quizrific are the student and the teacher. The functionalities associated with each role are presented in Figure 2. Both students and teachers need to login in order to use the Android application (see Figure 2 below).

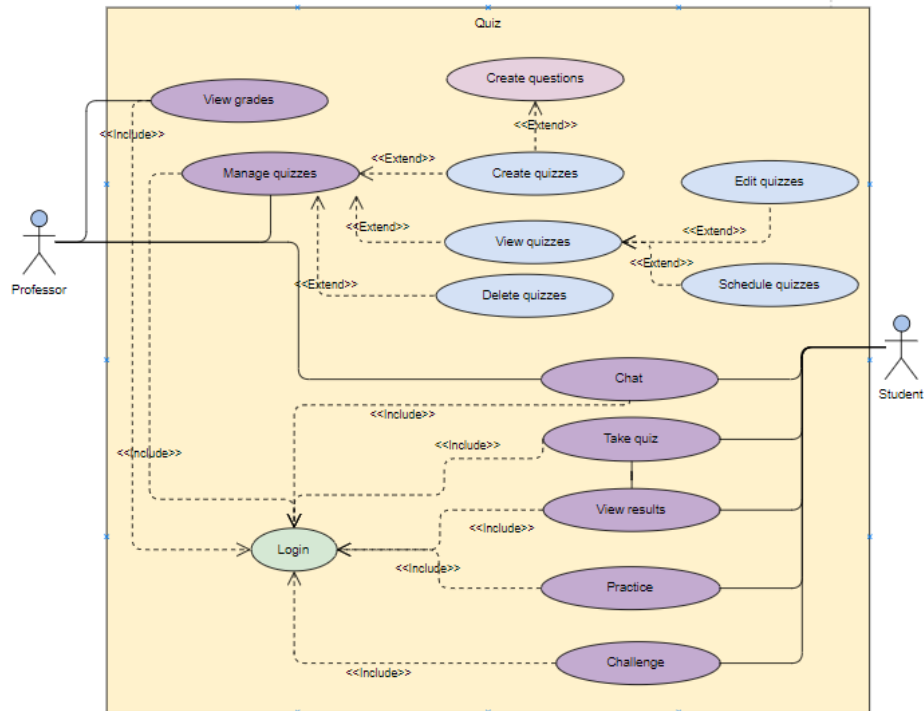


Figure 2. The software usecase diagram of Quizrific.

Similar to the case of the professors, with the help of Quizrific, students spend less time worrying while waiting for a result. Every correct answer is graded properly and added to the final grade, so that the scores are calculated rapidly and displayed accordingly, after finishing the quiz. Since every evaluation should rely on identifying the mistakes the student made and help him/her at achieving a better understanding of the material taught at school, the focus is shifted from plainly displaying the final grade to encouraging the student to go through the parts of the quiz where he/she responded incorrectly and review what the correct answer is.

As students tend to spend a significant amount of time on their phones, Quizrific comes as an useful tool that they can use anywhere, anytime they please, as long as they have Internet access. Students do not necessarily have to stay at their desk, in front of the computer to get ready for the next day at school. This flexible nature of the application is intended to allow students to even do a quick preparation on the bus on their way to school.

The menu of the student account contains is structured as follows: Home, Progress, Upcoming, Practice, Challenge. Students are able to see their progress (Figure 8). A pie chart has been implemented to show them distinctly how many correct and wrong answers they have given to a certain quiz. Additionally, there is another line chart in which their quiz results over a longer period of time are displayed, so the students can see where they are headed. In the Upcoming Fragment, learners can view a list of scheduled quizzes. The ones that were supposed to be taken at an earlier hour can no longer be accessed. Only when the current time corresponds to the one when the quiz is scheduled, the quiz can be taken. It is note-worthy that a student can miss the few first minutes and come later to take the test, but the timer will start with the remaining minutes of the quiz.

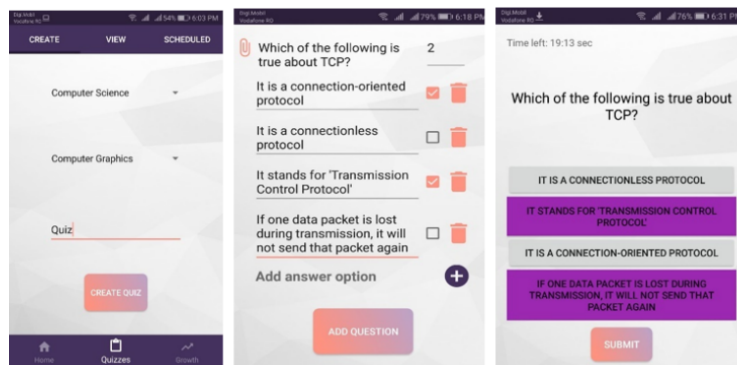


Figure 3. Multiple answers can be correct.

Informative charts and statistics are generated after every quiz, for the student to clearly assess his/her level of knowledge, get encouraged to improve it (or maintain the already-attained perfect score) and get a glimpse of how his/her efforts turn into real better results (see Figure 8). Test results are presented graphically as charts to the teacher, too (see Figure 7).

Aside from doing practice quizzes individually, a more fun way to prepare for school is forming the so-called group-study where students can help one another by discussing and sharing ideas. Quizrific incorporates both types of preparation methods. In the first manner, students have access to a set of quizzes proposed by their professors for them to practice on.

Individual testing

The workflow of practicing on individual questions is the following: the

student is shown one question followed by a set of answers, he/she chooses an answer and immediately sees if it is correct or wrong, keeps searching for the correct one and has the option to save the respective question for later reviewing. Navigating back and forth through the list of questions is also possible, if the student wants to work on a certain question at a later time. There is no time limit for this module, as the student is encouraged to take his/her time and vigorously assimilate the information.

Collaborative testing

For the collaborative method of learning, Quizrific offers a multiplayer module, called Challenge, where students can compete in real-time against one another in a simulated environment of a quiz (see Figure 4). A Challenge consists of two students connecting to the server and getting the same quiz containing a set of questions in the same order. Each question has multiple answer options of which only one is correct. Both students have a limited amount of time, i.e. 10 seconds, to choose the answer they think best suites the given question. After the timer is done, the correct answer along with all the players' answers are shown in a short timeframe, so that the players can learn what the accurate answer is and if the opponent has chosen it. For each correct answer, the player receives one point. If a wrong answer is given, or none at all, one point is subtracted from player's score, unless it is equal to zero. At the end of the quiz, the player with the biggest score wins (if both players acquired the same score, they are both equally declared winners). This competitive game-like quiz-preparation can consistently help at motivating students to improve their expertise on the specific subject and defeat their opponents.

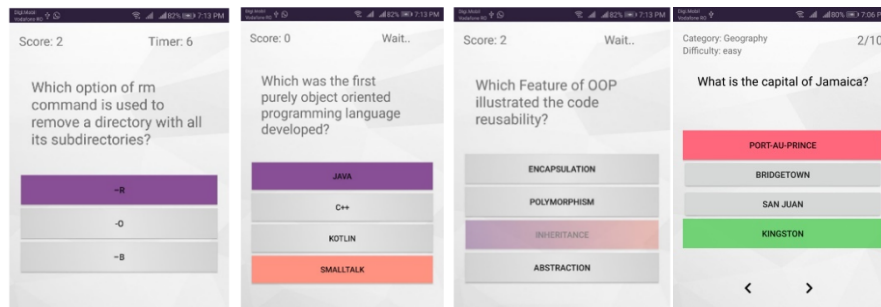


Figure 4. Multiplayer mode (first 3 screenshots) versus individual practice (the last screenshot).

Another feature Quizrific offers is that the professor can add images from which students to extract the correct answer(s). Future improvements include the option to add not only images, but also video, audio or pdf files through which more abilities like effective listening or reading comprehension could be assessed. Thereafter, the professor can view the list of created quizzes, schedule them, edit their contents, or even delete them (see Figure 5). All these are possible through the help of intuitive and modern user-interface design which makes use of user-friendly features provided by the Android SDK (Software Development Kit) such as swiping or scrolling gestures.

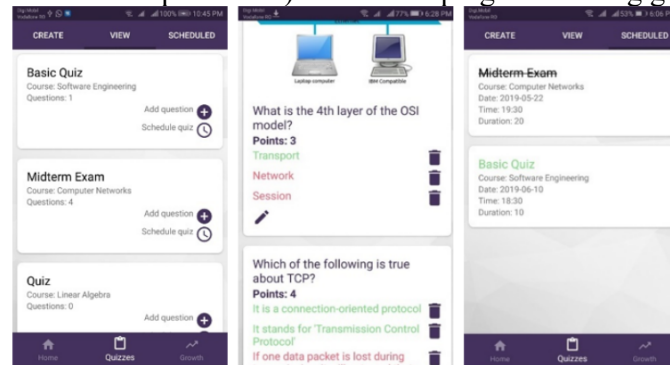


Figure 5. Viewing, editing, and scheduling quizzes.

To schedule a quiz, the professor has to pick an appropriate date, time and duration for the quiz to take place. Scheduled quizzes can be visualized in a separate tab, strikethrough text being used for past quizzes, green-colored text for upcoming ones, and green along with pulsating text for ongoing tests. The workflow of a timed quiz is described in Figure 6.

Once a quiz is done, i.e. the timer has stopped and students can no longer answer to questions, all their final grades are included in a pie-chart in which the professor is able to clearly visualize how many students have passed and how many failed – the passing grade considered five out of ten (see Figure 7).

In addition, a more detailed bar-chart can be extended to observe how many students got a certain grade. Last but not least, an individual line chart containing the results obtained at a certain course over a longer period of time provides the professor with the necessary information to analyze which student has improved drastically, who is better at a certain subject or who needs more help at grasping the notions presented at school.

Implementation details

Quizrific was developed in Android Studio; the graphical interface was designed with XML and the functionalities were programmed in Java. For storing users' details, quizzes, results, etc. we use MySQL as the database management system. In our case, relational was preferred over NoSQL, because integrity constraints were considered very important in the case of our app. PHP7 was used as the back-end server-scripting language needed to act as a bridge between the Android client and the database on the server (Android, September 2019). We also used the Volley library for processing HTTP requests. Volley is an HTTP library which was introduced by Google to make networking for Android applications easier and faster (Volley, September 2019). It integrates easily with any protocol and has support for raw strings, images, and JSON.

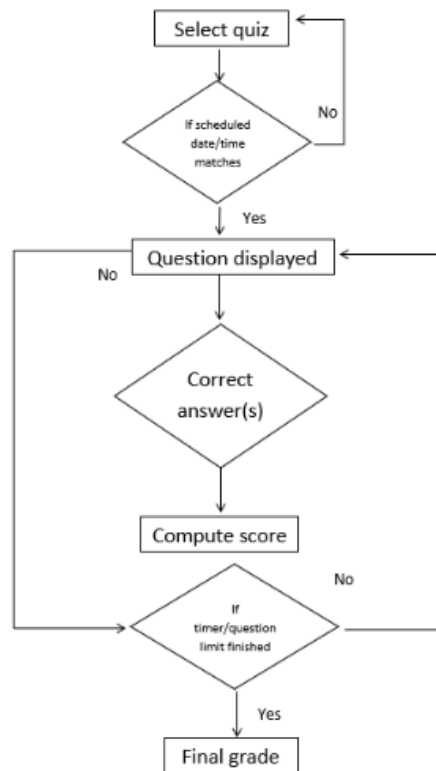


Figure 6. Workflow of the timed-quiz mode.

For the real-time functionalities – the communication system and the quiz competition – the web server was configured with `node.JS` (node.JS, September 2019). We used JavaScript and `Socket.io` (a JavaScript framework for web sockets) to code the dynamics of the application (Socket.IO September 2019). `Picasso` – an Android library created for managing images is used to handle automatically common inconveniences of loading images in Android (Picasso, September 2019). `MPAndroidChart` is a chart library used for data visualization, used for displaying statistics of test results (MPAndroidChart, September 2019).

4. Discussion

The target group for this application involves users in two categories: students and teachers. Among these, a representative sample should contain users of various age-groups, and various backgrounds with respect to technology usage. However, at this point, the app was tested on a sample of students from the Faculty of Computer Science and Mathematics, Ovidius University of Constanta. The procedure involved the students taking a quiz with questions from the Computer Networks discipline (Computer Science track, 3rd year of studies).

The results of the assessment are encouraging, the students considered the app useful for the purpose of learning. One expert teacher was asked to evaluate the system. She considered the app is task-oriented and displays consistent information, yet was intrigued by the numerous testing possibilities and suggested that the app provide several short films of its usage, together with an explanatory user guide. Further usability testing needs to be employed, in various scenarios and with more representative samples of users (Pribeanu, 1999).

The software described in this paper is freely available at the repository <https://github.com/krystal25/quizrific>.

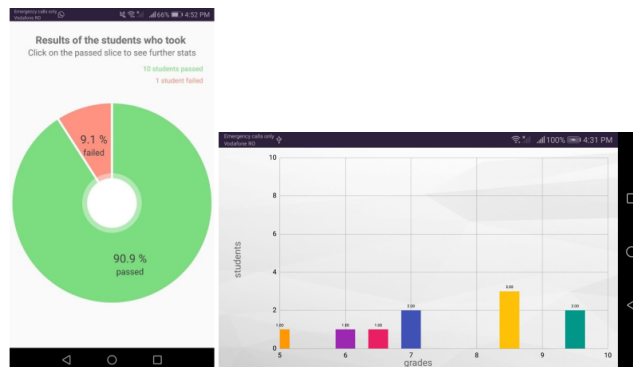


Figure 7. Charts for displaying statistics for quiz results for teachers.

A special feature that students can benefit from using Quiziriffic is the multiplayer challenge that allows them to study together by taking the same quiz and comparing their answers and scores. Preparing for a quiz is thought to be more enjoyable when forming a study-group where students can share their knowledge and empower one another.

Created for student preparation and evaluation, the attention might seem concentrated on the student, yet professors' role in the application is important, as they populate the app with quizzes that students can practice on or actually take at a certain date and hour. Quizirific fulfills professors' needs by bringing in one application the possibility to create a quiz, view the results and get in touch with their students in real-time. It automatically calculates the final grade, taking into account partial credit.

Students can easily prepare for different subjects, by taking practice quizzes or competing against each other in the simulated environment of a quiz. Statistics and charts are used to visually display test results. Although there exist other e-learning projects that test students' grasp of the studied material, Quizirific comes with its own contributions like: the multiplayer challenge system that allows students to play in real-time and treat quiz preparation as a game, the ability to communicate directly between students and professors (professors having the option to send important notifications to their students' phones) and all the other features incorporated in the same application.

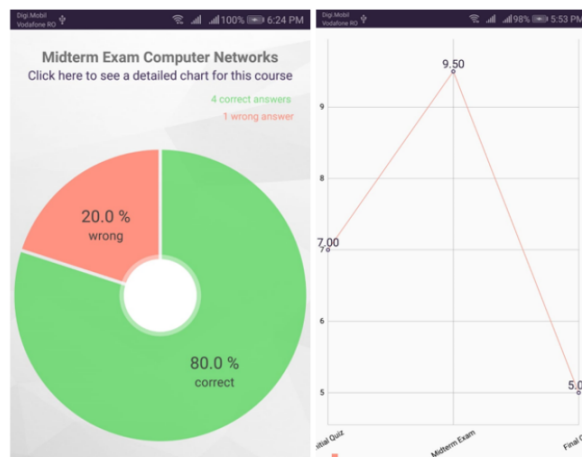


Figure 8. Statistics displayed for the student.

As with any project, there is room for continued improvement and growth. A custom grading algorithm based on the 'guessing penalty' (deducted points for each wrong answer given) could represent a future refinement. Another possible enhancement would be to allow the professor to add other types of files to a question, like video, audio or pdf files, not only images. This way,

more abilities like reading comprehension or effective listening can be tested. Leader boards could also be introduced to keep the past scores of the students that participated in challenges and motivate them to outgrow their peers. Similarly, a level-based progression would increase the hours invested by the students in the application to prepare for school, as the rewards by levelling up could consist of significant bonus points to their grades.

References

- Adam, D., Kioutsiouki, D., Karakostas, A., Demetriadis, S. (2014). "Do Your Students Get It? Quiz It! The Android Classroom Response System". *IEEE 14th International Conference on Advanced Learning Technologies*.
- Asghar, M.Z., Sana, I., Nasir, K., Iqbal, H., Kundi, F.M., Ismail, S. (2016). "Quizzes: Quiz Application Development Using Android-Based MIT APP Inventor Platform. In (IJACSA) *International Journal of Advanced Computer Science and Applications*, Vol. No. 5.
- Bautu, E. and Draghici, C. (2019) "Quizrific: an Android-based app for examinations", *RoCHI 2019 – International Conference on Human-Computer Interaction*.
- Edwards, J., Bagozzi, R. (2000) On the nature and direction of relationship between constructs and measures. *Psychological Methods* 5(2), 155-174.
- Krosnick, J. A. (2018) "Questionnaire design.", *The Palgrave Handbook of Survey Research*. Palgrave Macmillan, Cham. 439-455.
- Jiping, L., Xiao, H. and Zhang, H. (2019). "Design and Development of C++ Quiz System for Android-based Platform." *The ASEE Computers in Education (CoED) Journal* 10.1.
- Paternò, F. (1999) *Model based evaluation of interactive applications*. Springer Verlag.
- Pribeanu, C. (1999). "Un model detaliat al utilizabilitatii sistemelor interactive." *Revista Informatică Economică* 10: 31-36.
- Raymond, M. R., Stevens, C. and Bucak, S.D. (2019). "The optimal number of options for multiple-choice questions on high-stakes tests: application of a revised index for detecting nonfunctional distractors." *Advances in Health Sciences Education* 24.1: 141-150.
- Shanmugapriya, M., and Tamilarasi, A. (2011). "Designing an m-learning application for a ubiquitous learning environment in the android based mobile devices using web services." *Indian Journal of Computer Science and Engineering (IJCSE)* 2.1: 22-30.
- Android (September 2019), Official Android documentation, retrieved from <https://developer.android.com/guide>
- Kahoot (Septembrie 2019), <https://kahoot.com/b/>
- MPAndroidChart (September 2019), Official MPAndroidChart external library documentation, retrieved from <https://github.com/PhilJay/MPAndroidChart>
- Node.JS (September 2019), Official node.JS documentation, retrieved from <https://nodejs.org/en/docs/>

QuizUp (Septembrie 2019), <https://www.quizup.com/en>

Picasso (September 2019), Official Picasso library documentation, retrieved from <https://square.github.io/picasso/>

PhP (September 2019), Official documentation for PHP, retrieved from <https://www.php.net/docs.php>

Socket.io (September 2019), Official documentation for socket.io library, retrieved from <https://socket.io/docs/>

Volley (September 2019), Official documentation for Volley library, retrieved from <https://developer.android.com/training/volley>.