

# ***UML Quiz*: AUTOMATIC CONVERSION OF WEB-BASED E-LEARNING CONTENT IN MOBILE APPLICATIONS**

Alexander von Franqué

*Vienna University of Technology*

*Favoritenstrasse 9-11/187, A-1040 Vienna, Austria*

Hilda Tellioglu

*Vienna University of Technology*

*Favoritenstrasse 9-11/187, A-1040 Vienna, Austria*

## **ABSTRACT**

Many educational institutions use Learning Management Systems to provide e-learning content to their students. This often includes quizzes that can help students to prepare for exams. However, the content is usually web-optimized and not very usable on mobile devices. In this work a native mobile application (*UML Quiz*) that imports quiz content from Moodle and adapts it for a mobile environment has been developed. The result is highly usable and promising. The existing teaching content can be easily used in the new system by offering advantages of mobile applications. In this paper, the design decisions for *UML Quiz* and the usability optimizations to the content are presented. The results are evaluated and further possible improvements are discussed.

## **KEYWORDS**

E-learning, m-learning, mobile, usability, native, automatic content adaption, automatic content conversion.

## **1. INTRODUCTION**

Many educational institutions use web-based Learning Management Systems (LMS) to provide e-learning content to their students. Popular examples include Moodle, Blackboard, Desire2Learn, and ILIAS. Besides being a well-known and mostly used system Moodle “is not intrinsically designed to be accessed through mobile devices, but the constructivism pedagogy supported in Moodle VLE could be applied to design the interaction in m-learning” (Hewagamage et al., 2012, p.2). Moodle does not fully support JavaScript. This makes it difficult to access via mobile devices.

Most of the LMS platforms include quiz modules. These quizzes can, for example, help students to prepare for exams. However, these platforms rarely offer easy access for mobile devices, especially not with native applications. Native applications offer numerous (usability) advantages over web-based apps, especially if they are running on mobile devices. However, the creation of native apps is time-consuming and expensive.

In this work we propose a native mobile quiz application (*UML Quiz*) that can deal with web-optimized e-learning content in a way that preserves usability and encourages easier learning. It is accompanied with an automatic system that converts the web-optimized data into a mobile-friendly format, both in size and layout. A few optimizations and enhancements to the data are being used to make the application compelling and easy to use, even though the data was not originally prepared for this specific device.

*UML Quiz* has been developed natively for iPhone and iPad to use for teaching at Vienna University of Technology (TU). The course supported by *UML Quiz* is about basics of the Unified Modeling Language (UML) to students of computer science and business informatics. In this course, a modified version of Moodle (TUWEL) is being used to display quizzes to the students, which they can use to prepare for exams. These quizzes include lots of diagrams of different sizes and very long questions and answers. The system we propose imports the existing, web-optimized data from TUWEL and deals gracefully with these types of

questions. It allows the university to keep using the existing data and displays it in a highly usable manner on a mobile screen.

In the next section we briefly introduce relevant related work before we analyze *UML Quiz* in this research scope. We also will touch upon other work that adapts web-content for mobile devices. Section 3 presents implementation details of *UML Quiz* with a focus on design and usability decisions. Section 4 shows the evaluation of the results, which we further explored for future work.

## 2. RELATED WORK

In this section, we want to show some other systems developed so far. Second, we want to focus on usability and content conversion in such systems. We will also highlight the differences to our system *UML Quiz*.

Despite the fact that “accessing learning content through mobile devices is not very comfortable” (Hewagamage et al., 2012, p.32) and e-learning content can be accessed the best through desktop interfaces, the advantages of mobile applications especially for learning cannot be underestimated. There are several systems that are implemented as mobile applications, mainly quizzes for teaching.

The group of Schön et al. (2012a, 2012b) developed a mobile quiz application for the University of Mannheim. Its main purpose is capturing classroom feedback from the students during courses, which is the main difference to *UML Quiz* that mainly assists students in the preparation process for exams. It is a web-based application and supports dealing with multiple- and single-choice questions. The application’s data is formatted to fit the mobile screen. Question texts are kept short, possible answers are usually single words, and images are kept at a size which are clearly visible on a mobile screen. Another system was developed to support learning foreign languages (Tabata et al., 2010). It uses the same approach as the one from the University of Mannheim. There are text-based questions and answers whereas texts are kept very short. The system supports only multiple-choice questions.

iClass is a further system for mobile learning (Eyhab and Qusay, 2012). In contrary to the systems above, it hasn’t been developed on top of an existing university infrastructure, but instead has been designed for mobile devices from scratch. However, it is a web-based implementation using the phone’s browser, and this is the main difference to *UML Quiz*. The decision for this technology was made for the reason to support a variety of devices. Authors did not want to “enforce a particular device” (Eyhab and Qusay, 2012, p.591). At the same time, they are aware of the reduction of usability because of not taking care of the platform and the follow up limitations in the interaction with the system.

The multiple-choice questions realized in (Montanaro, 2012) resemble closely to the structure of *UML Quiz*. The content presented is dynamic, the text can be displayed in multiple rows, and feedbacks are being shown after having solved the question.

*UML Quiz* closely follows the general structure of most of these systems, and in some aspects improves upon them. One of the main differences is that *UML Quiz* is implemented natively, whereas most of these systems are web-based. We found the following systems as native mobile applications that we unfortunately could not test because of the license restrictions: Upside2Go with a quiz module, Mooestro without a quiz module by focusing on orientation at campus and while registering for lectures, Pearson Learning Studio and OpenClass as a very popular LMS, MOBL21 including multimedia to its quiz module, which is the closest to our *UML Quiz* among all. Furthermore we found Adobe Captivate and Articulate as HTML5-based systems, again with no open access. Other systems available like Quiz and Flashcard Maker, Edmodo, Blackboard Mobile, or Socrative are not described here because of the lack of usability or features they offer.

The second important characteristic of *UML Quiz* is the focus on usability and easiness. Even though the most of the mentioned systems have been developed with the aim to support mainly mobile platforms (Wasserman et al., 2010), they are not exhaustively using the possibilities of a mobile device. This is only possible if the system is developed natively for specific devices, what we do with *UML Quiz*.

KnowledgePulse (Bruck et al., 2012), Quizzer (Giemza et al., 2012) and MLEA (Castillo et al., 2012) are also native applications for iOS and Android as *UML Quiz*. They also constrain the content of the data to fit certain screen size of mobile devices. This scalability feature is a limitation of native applications (Giemza et al., 2012) and a challenge to expand the system to another platform. *UML Quiz* would have to be completely re-written in Java to work, for example, on Android.

Contrary to scalability, usability is an important advantage of native applications. Native applications usually perform much faster than web-apps – especially scrolling is much more responsive (Charland and Leroux, 2011). Native apps enable access to device-specific APIs, like Apple’s Core Data, which has been used in the development of *UML Quiz*. They allow for offline use, which is particularly useful in the context of mobile learning. However, web-apps have the advantage of being platform independent and the development costs are usually much lower than the cost of building native apps (Selvarajah et al., 2013).

There are other systems adapting web-content for mobile devices (Casany et al., 2012), some specifically for e-learning material (Gutierrez-Martinez et al., 2012) However, these works concentrate mainly on the format conversion and the interoperability between different learning management systems, but they do not take the usability implications into consideration.

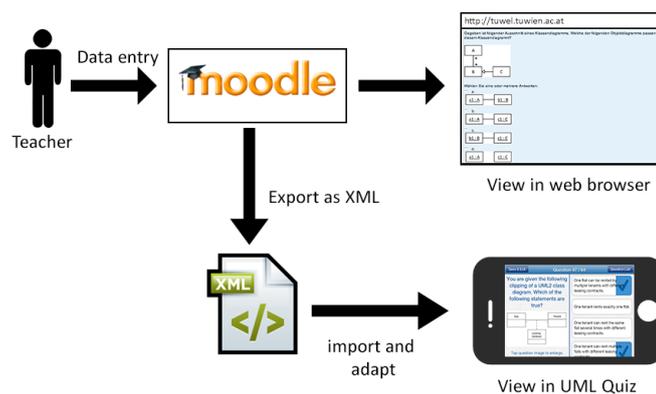
Romero et al. claim that when creating m-learning systems “designers of both software and content have to consider the special restrictions associated to their use” (2010, p. 375). The limiting factors are among others screen size, resolution, and special input mechanisms. Guidelines for designing usable educational material suggest that the display area should not exceed the screen size and the content should be segmented into smaller chunks. Additionally extensive scrolling and the number of clicks should be well designed and one should pay attention on the amount of relevant information on each page. Further points of importance are highlighted like clear navigation structures, consistency of interface elements, and typeface.

### 3. *UML QUIZ*: CURRENT IMPLEMENTATION

*UML Quiz* was designed to automatically import the web-based quizzes from Moodle and display them in a highly usable manner on a mobile screen. However, as Moodle supports free HTML formatting of the content (text of any length and any number of images), constraints and design goals had to be defined. We did not want to restrict the length of the question or answer text, as the system should be able to deal with existing content. However, we constrained the use of images to one image per question and one image per possible answer. *UML Quiz* automatically discards questions that do not follow these constraints. We also wanted to ensure that navigation between questions is fast and easy. Romero et al. (2010) prefer paging over scrolling when navigating, so in *UML Quiz* each question consists of a screen-full content. We implemented a simple swipe gesture for navigation between questions. The overall design is focused on providing as much space for the content as possible and minimizing the space that UI-elements take up. We wanted to make sure that most elements of a question (question text, images, and possible answers) are visible at the same time.

Our system uses Moodle’s export feature to obtain quiz data in the Moodle XML format. We then parse the data and import it into our native mobile application, *UML Quiz*. Upon import, the content is optimized for use on mobile devices. See Figure 1 for an architectural overview.

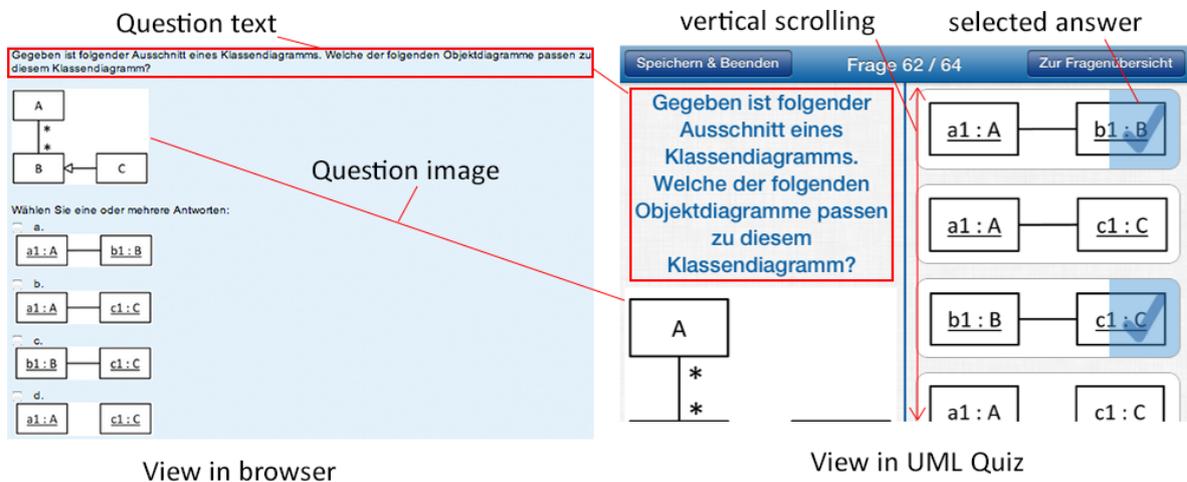
Figure 1. Architectural overview.



In TUWEL (a modified version of Moodle at Vienna University of Technology) different question types are presented to the user. The text in all types can be formatted freely with standard HTML tags such as bold and italics. Any number of images can also be added at any point. Each question type has been individually adapted for use in the native app. The question and possible answers are presented as HTML-formatted text

to the user. Usually, questions contain at most one image in the question text and one image per answer. However, in TUWEL, there are no constraints to enforce this rule, and theoretically more images are possible. See Figure 2 for an example, which illustrates how this is solved in TUWEL and implemented in *UML Quiz*.

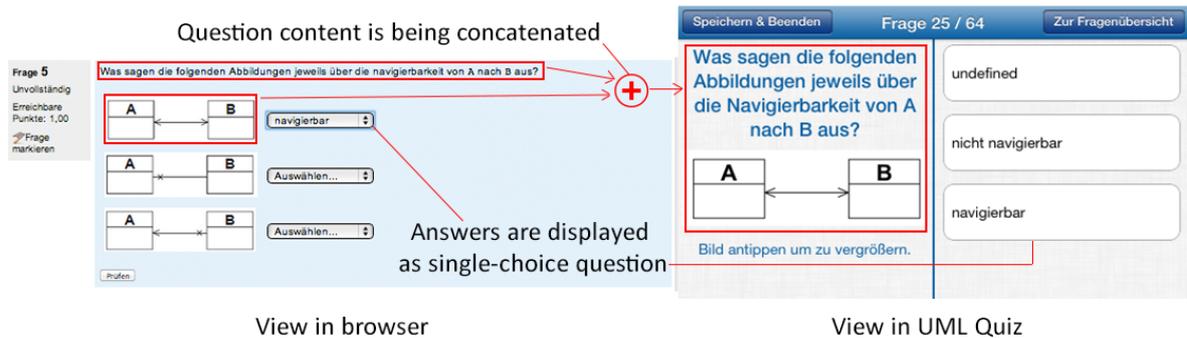
Figure 2. Multiple-choice questions in Moodle and *UML Quiz*.



The Figure 2 also shows the same question after it has been imported into the mobile application. The interface is forced into landscape mode, splitting the screen's content into half. On the left side, the question text and, if available, a single question image are displayed. Tapping the image enlarges it to full screen, also enabling zooming and panning with two-finger pinch gestures. On the right side, the possible answers are displayed, some of which are selected in Figure 2, which is visualized with blue checkmarks. Both sides of the screen are individually scrollable vertically, if the text is longer than screen size allows for.

In TUWEL, some questions are implemented with a dropdown list. In these questions, a general text that explains the task is presented to the user (e.g., "Select the right answer from the list"). Then, a number of subquestions is being shown. For each of these subquestions, a drop-down list is displayed. The possible choices of this list are the same for all subquestions. See Figure 3 for an example.

Figure 3. Optimization of drop-down questions.

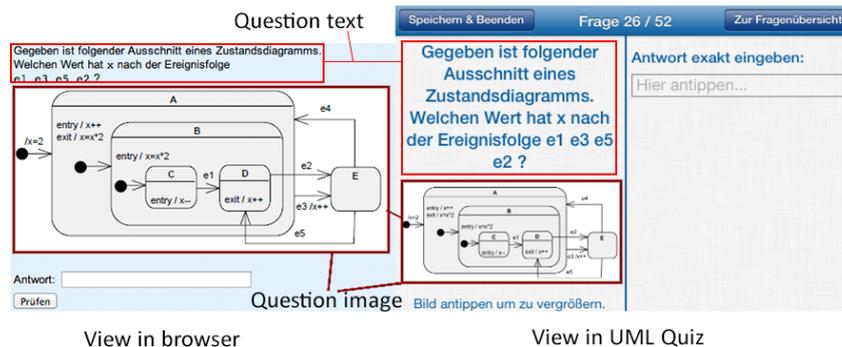


On touch-screen devices drop-down lists are usually implemented with picker control elements. Early iterations of *UML Quiz* have implemented this control element to support drop-down questions, however, the picker takes up so much screen space, that all the other content on the screen is occluded. To solve this problem, the following optimization has been used. Upon importing the TUWEL data, questions of the drop-down type are split into numerous single choice questions. Figure 3 shows a drop-down question with images in TUWEL and compares it with the first part of the same question split into a single choice question in *UML Quiz*. We interpret the possible selections from the drop-down list as possible answers. The question text is constructed by concatenating the general instruction text of the drop-down question with the text or image from the specific question.

While this approach works much better than the use of the picker control element, it doesn't quite fit some of the existing data. In the TUWEL screenshot above, the general text is written in plural, as it refers to multiple images, however, in *UML Quiz*, only one image is visible. In this case the consequences of this issue are harmless, as it merely might confuse users. However, in other cases, this problem might completely transform the original sense of the question. As this limiting problem is caused by semantics, it is not possible to solve it in an automatic fashion; it requires a manual re-work of the questions.

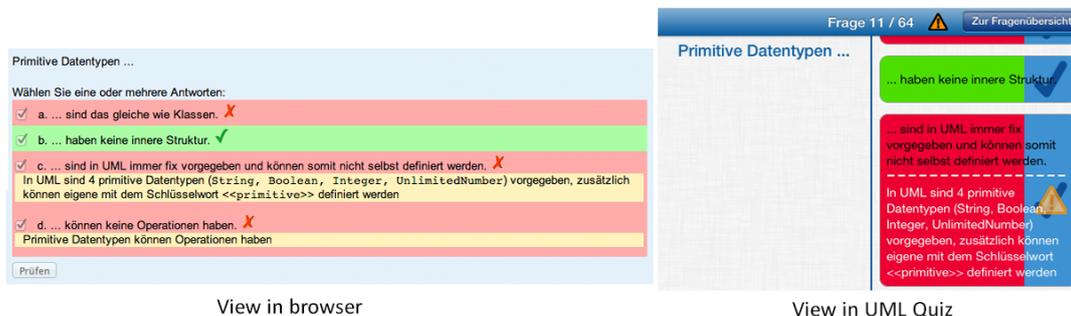
Another question type is presented with a text box in TUWEL. These questions do not offer users a list of possible answers to choose from. Instead, they are prompted to enter the answer in a text box. The question itself is presented to the user with text, an image, or both, and usually requires them to calculate a number or a certain answer. See Figure 4 for an example and a comparison with the mobile application. As visible in the screenshot, a simple text field has replaced the left side of the screen. Tapping it reveals the on-screen keyboard, as well as a button to dismiss it again.

Figure 4. Comparison of text-box questions.



After a question has been solved, its correct answers are revealed (see Figure 5). Another feature of the system is the explanation text that is displayed to users after they finish a quiz. Its purpose is to provide additional information about the question or the possible answers. The text may also explain why certain answers are correct or incorrect. It is also possible to include images in the explanation.

Figure 5. Comparison of solved questions.



*UML Quiz* offers some additional features that help improve the usability of the system:

- Flashcard system – The application remembers how often each question has been solved and thus allows the user to practice depending on progress.
- Persistent quiz data and progress storage – In the final application, quiz data is loaded on first launch from XML files and then imported in a more convenient format. For access, Apple's persistence framework, Core Data, was used, which uses an SQLite database as underlying storage format. All data from the XML files is stored, and is associated with some new data to track the user's progress. The application remembers how often each question has been answered, which enables the flashcard system to divide up the questions into boxes. Also, the progress of saved, unfinished quizzes is stored. In the application settings, users also have the option to completely reset their progress.

- Multiple language support – The application loads different XML files depending on the device language. Quiz progress is stored independently for each language: whenever a new language is loaded, a new SQLite file will be created. When users switch languages, they are prompted if they want to keep the old language database or switch to the new one.
- Online question updates – It is possible to download new XML files from within the application. It will then parse them and update, add or remove any new content. It is also possible to add new images this way. They are first compared by timestamp, so that files that haven't been changed won't be re-downloaded this way. This feature enables teachers to change questions after the application has been shipped.

## 4. EVALUATION

UML Quiz was tested by 6 persons<sup>1</sup> from different age groups and with various educational backgrounds. Some were interviewed multiple times during the development process, and some were only shown the final application.

- Multiple iterations of *UML Quiz* were shown to Ulrike (55 years old, college dropout), including the final application. First prototypes of the app were in portrait orientation and had the entire content of a question (question text, answers, images) scroll vertically. Ulrike did not like this interface at all: solving a question was a very tedious process for her, as she had to scroll constantly between the question text and possible answer choices. To fit more content on the screen, a small text size was chosen, which she also criticized, since it was hard for her to read. In the final application, she said that most of the issues had been improved: the landscape layout made it easier for her to solve the questions, as most of the content was visible at the same time, requiring less scrolling. Font sizes were larger and the navigation between questions was much clearer.
- The full application was shown to Micheline (28 years old, non computer science graduate student) after most parts were completed, not having seen the quiz in TUWEL before. She was initially confused with the navigation, not understanding the horizontal swipe gesture to navigate between questions. However, when she had grasped the feature, she stated that she liked it. She was also initially confused by the colors that were used to mark solved questions as correct or incorrect, thinking that the blue check signified correctness, not selection.
- Lukas (23 years old, computer science undergrad student) was shown the full application after most parts were completed. He had seen and used the quiz in TUWEL before, as he participated in the UML course himself. He positively noted the navigation between individual questions and he really liked that one could see all the elements of a question in a single screen. The question overview button at the top left corner of the screen initially confused him, because he wasn't sure where it would lead him. However, when he pressed it and then navigated back to the original question, he understood what had happened. He also stated that he would have used the application, had it been available at the time he took the course.
- Moritz (27 years old, computer science undergrad student) was also a participant in the UML course and had also seen the questions beforehand in TUWEL. He was also shown multiple iterations of the application, including the final result. He did not like the navigation between questions and was especially confused by the button leading to the question overview. The main reason for his dislike was the problem caused by both horizontal and vertical scrolling, as described in the previous section. He positively noted the flashcard system and the online updates.
- Wolfgang (57 years old, non computer science graduate, working as a high school math teacher) is without a strong technological background. He liked the navigational layout and said that it was easy to understand and use. He especially noted the overview screen, allowing to see which questions had been answered correctly after finishing the quiz. However, he thought that the font sizes were sometimes too small and hard for him to read. He liked that the question images were zoomable. He disliked that the answer images are not zoomable, again making it harder for him to

---

<sup>1</sup> We thank Marion Scholz for her ideas, support, and supervision during the creation and establishment of the *UML Quiz* and all students who evaluated the system and spent time with us to give their valuable feedback.

read. From his viewpoint as a teacher, he liked that the content could be revised and adapted through online updates, making it easy to add or change questions.

- Marion (instructor of the UML teaching course at Vienna University of Technology) accompanied the entire development process, providing valuable feedback from the perspective of an instructor. She also did not like the initial prototype very much, as the portrait layout made it hard to solve a question without scrolling. She appreciated the switch to landscape mode and suggested further improvements to the layout, which included increasing the font size and removing some UI elements. The initial prototype had an additional UI element to jump from question to question, which was removed in favor of more space for actual content. Marion also liked the idea of splitting drop-down questions into multiple individual single-choice questions, however she was concerned that it might cause confusion for people entering the data in Moodle, who are not familiar with the mobile app.

These tests have shown that *UML Quiz* improves the usability of the quiz enormously, when compared with the access on a mobile browser on Moodle. The big advantage is that most parts of a question are visible in a single screen and scrolling can be avoided in most cases. Users also reacted very positively to the additional features described in the previous section, especially to the flashcard system. The system generally does a good job in converting the web-based content and making it usable on the mobile screen.

However, *UML Quiz* still has some usability issues, which need to be solved in future work. Most of them are contextual problems caused by the content used in the quiz. Such content has usually been designed with and for desktop screens. An example is the way that *UML Quiz* deals with images. Because most images in the existing data are rather small, *UML Quiz* scales them to the largest size possible with the given screen space, while maintaining aspect ratio. While this works well with most images, it sometimes produces unwanted results, which leads to images like in Figure 6a. In the screenshot, a giant element from a UML diagram is visible, even though the element's actual size is only a few pixels in diameter.

Figure 6. a) Very large images. b) Small unreadable images. c) Long question text. d) keyboard obscuring content.



On the other hand, some images are much too small to read on a mobile device (see Figure 6b). Tapping the image enlarges it to full-screen and enables zooming. However, in full-screen mode it is not possible to see the question text and possible answers at the same time. Thus, users are again required to switch back and forth. In this design, tap-to-zoom could not be provided to enlarge the images displayed as answers, as the action of tapping already selects and de-selects the answer.

Students also complained about still having to scroll too much, which is the consequence of too long question texts or too many answer choices (see Figure 6c). Another problem was caused by both sides of the screen being scrollable vertically, as well as horizontally: Sometimes users try to scroll horizontally to get to the next or previous question, but instead the gesture is recognized as a vertical scroll by the application, and vice versa. This caused a lot of frustration.

Some users criticized the text-box questions because the on-screen keyboard takes up too much space, obscuring the content (see Figure 6d). Confusion was also caused by the problems arising from splitting drop-down questions into multiple-choice questions, as described in the previous section.

Further improvement of the content could solve most of these problems. However, it will likely require manual adjustment of the data. Possible improvements can target data input, as the input mask in the Moodle quiz module is not intended to produce mobile-friendly data.

## 4. CONCLUSION

In this paper we have shown how a native mobile application can help to improve the preparation for exams at university level. Based on the assumption that “m-learning extensions cannot be developed to an existing e-learning system without changing the pedagogy and design of the learning content” (Hewagamage et al., 2012, p.13), we designed and developed the native mobile app *UML Quiz* and evaluated from different points of view. The results of this evaluation show that it is possible to take existing quiz data from web-based LMS and adapt it for use on mobile devices. Despite the fact that there are several areas for improvement both on conceptual and implementation level, students overall have given very positive feedback to *UML Quiz*.

Further research to improve the content can aim the creation of the content. Another possibility would be a semi-automatic approach to improve the content. Mockups for a new input mask exploring this idea have already been created. The pre-existing content is imported and presented in a live preview in actual size of the mobile screen. It helps the teachers to judge if the content fits the screen and, if necessary, to make adjustments to it (for example, shorten the question text). This approach has received a good response in first evaluations with teachers.

## REFERENCES

- Eyhab, A-M. and Qusay, H. M. 2012. iClass: Improving the Mobile Learning Experience. *Proceedings of ICCIT 2012*, pp. 587-592.
- Bruck, P. A. et al., 2012. Mobile Learning with Micro-content: A Framework and Evaluation. *Proceedings of 25<sup>th</sup> Bled eConference eDependability: Reliable and Trustworthy eStructures, eProcesses, eOperations and eServices for the Future*. Bled, Slovenia, pp. 527-543.
- Casany, M. J. et al., 2012. Extending Moodle Services to Mobile Devices: The Moodbile Project. *Proceedings of UBICOMM 2012: The Sixth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*. Barcelona, Spain, pp. 24-28.
- Castillo A. et al., 2012. A System for Mobile Learning: A Need in a Moving World. *Procedia - Social and Behavioral Sciences*, Vol. 83, pp. 819-824.
- Charland A. and Leroux, B. 2011. Mobile application development: web vs. native. *Communications of the ACM*, Vol. 45, No. 5, pp. 49-53.
- Giemza, A. et al., 2012. Challenges in Scaling Mobile Learning Applications: The Example of Quizzer. *Proceedings of 2012 IEEE Seventh International Conference on Wireless, Mobile and Ubiquitous Technology in Education (WMUTE)*. Takamatsu, Japan, pp. 287-291.
- Gutierrez-Martinez J. et al. 2012. E-Learning Contents Automatic Conversion. *International Journal of Innovative Computing, Information and Control*, Vol. 8, No. 3(B), pp. 2065-2088.
- Hewagamage, K. P. et al. 2012. “M-Learning Not an Extension of E-Learning”: Based on Case Study of Moodle VLE. *International Journal of Mobile and Blended Learning (IJMBL)*, Vol. 4, No. 4, pp. 21-33.
- Montanaro, A. 2012. *Implementing Mini Quizzes to Increase Student Learning for a Class via a Mobile Learning Application*. A Capstone Project Submitted to the University of North Carolina Wilmington in Partial Fulfillment of the Requirements for the Degree of Master of Science.
- Romero, R. et al. 2010. Designing Usable Educational Material for English Courses Supported by Mobile Devices. *Lecture Notes in Computer Science*, Vol. 6389, pp. 373-383.
- Schön, D. et al, 2012a. A lightweight mobile quiz application with support for multimedia content. *Proceedings of 2012 International Conference on e-Learning and e-Technologies in Education (ICEEE)*. Lodz, Poland, pp. 134-139.
- Schön, D. et al. 2012b. Integrating a Lightweight Mobile Quiz on Mobile Devices into the Existing University Infrastructure. *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2012*. Denver, USA, pp. 1901-1907.
- Selvarajah, K. et al. 2013. Native Apps versus Web Apps: Which Is Best for Healthcare Applications? *Lecture Notes in Computer Science*, Vol. 8005, pp. 189-196.
- Tabata, Y. et al. 2010. An iPhone quiz system for learning foreign languages. *Proceedings of 2010 2<sup>nd</sup> International Asia Conference on Informatics in Control, Automation and Robotics (CAR)*. Wuhan, China, pp. 299-302.
- Wasserman, A. I. et al. 2010. Software engineering issues for mobile application development. *Proceedings of the FSE/SDP workshop on Future of software engineering research*. Santa Fe, USA, pp. 379-400.