



An Analysis of *ProjectEdu*: A Mobile Learning Application for Software Project Management Education

Maria Lydia Fioravanti^(✉), Raul Donaire Gonçalves Oliveira,
Gustavo Martins Nunes Avellar, Camila Dias de Oliveira,
and Ellen Francine Barbosa

Institute of Mathematics and Computer Science (ICMC),
University of São Paulo (USP), São Carlos, SP, Brazil
{mlfioravanti,raul.oliveira,gustavo.avellar,camila.oliveira}@usp.br,
francine@icmc.usp.br

Abstract. Learning applications play a key role in educational activities, both in academia and in industry. In this scenario, mobile learning has emerged as a new and promising learning modality, providing more attractiveness, agility, interactivity and flexibility to learners, tutors and teachers in carrying out educational activities and practices. In a different but related perspective, according to ACM/IEEE curricula, Software Project Management is an important topic to be taught in Computing degree programs. Despite its relevance, frequently, there is no specific course to teach Project Management, so its concepts are taught along with the other concepts of Software Engineering. However, even when there is a course dedicated to the subject, in its traditional format, the subject of Project Management is approached in a theoretical way. In this context, it is important to seek strategies that motivate the teaching-learning process, such as, for instance, mobile learning. In order to provide a more interesting approach to learn Software Project Management, we designed, prototyped and evaluated a tool entitled *ProjectEdu*. In general, users were enthusiastic and positive about the use of the mobile learning application, but they also pointed out some improvement points to make the tool more attractive.

Keywords: Mobile learning · Project management education · Usability Test

1 Introduction

Learning applications play a key role in educational activities, both in academia and in industry [10, 26]. The miniaturization of electronic components and their cheapening have allowed the development of devices with processing capacity and functionalities equivalent or superior to those of many computers [30]. These changes associated with ubiquitous computing have leveraged a new modality

of learning called mobile learning (m-learning) [11, 15, 27, 29]. In this scenario, mobile learning has emerged as a new and promising learning modality, providing more interactivity and flexibility to learners, tutors and teachers in carrying out educational activities and practices [14].

As with many emerging paradigms, there are several attempts to define m-learning. However, it is noteworthy that, regardless of the various definitions over the years [11, 12, 15, 16, 20, 21], there is a convergence in definitions regarding the use of mobile devices to promote learning anytime, anywhere. Based on such definitions, we have adopted the following definition for this work:

“Mobile learning is a learning modality characterized by the ability to provide an effective interaction among users (learner, teachers and tutors), allowing them to contribute, participate and access the educational environment through mobile devices (cell phones, PDAs, smartphones, tablets, laptops, and so forth) anytime, anywhere.”

Portable technologies, together with computational networks as well as the dissemination and easy access to the Internet, are becoming more and more present in the daily life, promoting access to information in an easy and fast way [6]. This scenario has favored the emergence of new learning modalities, providing new means to address the deficiencies of traditional teaching, making it more agile, flexible and attractive [17].

When such technologies are used for educational purposes, they can promote an improvement in student learning and become pedagogical support for the teacher [28]. New technologies and teaching techniques, as well as current studies on learning processes, can provide more effective resources to meet and motivate those involved in the teaching and learning processes.

The challenges associated with mobile learning have been investigated and several supporting mechanisms have been proposed to assist in the design and evaluation of mobile learning applications. For instance, a pedagogical pattern language, namely *MLearning-PL* [13], and a requirements catalog, namely *ReqML-Catalog* [23].

In a different but related perspective, *Association for Computing Machinery* (ACM) and *Computer Society of the Institute for Electrical and Electronic Engineers* (IEEE-CS) have been involved in initiatives to develop curricular guidelines for typical Computing degree programs (such as Computer Engineering, Computer Science, Information Systems and Software Engineering). Among the guidelines, a body of knowledge was identified, organized hierarchically in areas, units and topics of knowledge to each program.

Following the structure of areas, units and topics proposed by ACM/IEEE to the Computer Science Curricula (version CS2013 [2]), Software Project Management unit is part of the Software Engineering area and must be addressed. Similarly, curriculum recommendations of other Computing undergraduate degree programs also include Project Management topics, such as Information Systems [3], Computer Engineering [1], Information Technology [4], and Software Engineering [5].

Despite its relevance, frequently, Software Project Management is approached in a theoretical way. In this context, it is important to seek strategies that motivate the teaching-learning process, for instance, mobile learning.

In order to provide a more attractive approach to learn Software Project Management, we designed a tool entitled *ProjectEdu*. The idea was to use the aforementioned artifacts and investigate whether the learners remained more motivated and committed to using the mobile learning app.

Considering this scenario, in this paper we evaluated *ProjectEdu*, a mobile learning application for Software Project Management education. The research question we aimed to answer is: “*What do users of mobile learning applications expect in order to keep themselves motivated and committed to using such applications, considering their different learning styles and needs?*”. In general, users were enthusiastic and positive about the use of the mobile learning application, but they also pointed out some improvement points to make the tool more attractive.

The remainder of the paper is organized as follows. In Sect. 2, we briefly present some studies about the supporting mechanisms used to design *ProjectEdu*. In Sect. 3, we discuss *ProjectEdu* and its design process. In Sect. 4, we present the evaluation methods used and discuss the results. Finally, we draw conclusions and provide directions of future work in Sect. 5.

2 Background

When dealing with domain-specific software, such as learning applications, we must be concerned about domain requirements, which are derived from the application domain of the system [24]. On the other hand, we must be concerned with specific needs and opinions of the end users.

We designed *ProjectEdu* aiming at a more attractive and motivating application. To achieve this goal, we used two main artifacts: *ReqML-Catalog* and *MLearning-PL*.

ReqML-Catalog is a requirements catalog for mobile learning applications. The proposition of *ReqML-Catalog* was motivated by a scenario where there was no complete and well-defined set of requirements for mobile learning applications. Aiming to bridge this gap, the work of Soad et al. [23] intended to be a step forward in this direction.

The categories defined in the catalog are divided into 12 requirements subcategories. Three subcategories are defined for the *Pedagogical* category. The first is *Learning*, which is defined by the application’s ability to provide features that contribute to student learning. Additionally, *Content* is defined by the ability to deliver manageable and quality content and *Interactivity* is defined as the ability of the application to provide features that help users interact with each other and with the application.

The *Social* category comprises *Socioeconomic* and *Sociocultural* subcategories. Finally, the *Technical* category is subdivided into *Functional Suitability*, *Performance Efficiency*, *Compatibility*, *Usability*, *Reliability*, *Security* and *Portability*.

In a related perspective, *MLearning-PL* [13] is a pedagogical pattern language for mobile learning applications, comprised of 14 patterns. The main audience of *MLearning-PL* is novice educators who occasionally must play a requirements analyst role in a mobile learning application project. Those educators can be benefited from *MLearning-PL*, once they can reuse pedagogical knowledge from senior educators.

It aims to assist in the definition of mobile applications for keeping learners motivated and committed to using such applications, according to their different learning styles and an effective knowledge acquisition. *Let's Play* [13], for instance, is a pattern which suggests to add games elements to the learning process to make learning fun.

Such artifacts are complementary and can be applied together in the process of defining a mobile learning application.

3 Overview of *ProjectEdu*

ProjectEdu is a mobile learning application prototype focused on users who want to learn Software Project Management. Several mobile applications can be found in order to carry out the management activities throughout the project, but *ProjectEdu* stands out since it focuses on teaching Project Management theory as well as its practice.

ProjectEdu is in its prototype version, being developed using Justinmind¹ tool. Figure 1(a) and (b) show some of the first screens the user will be in touch with: the main screen and login screen.

In its current version, *ProjectEdu* has the following main features:

- Activities:** In this area of the app, the learner has access to all the theoretical content of Software Project Management and also to some related activities and practices.
- Statistics:** This feature allows the learner to check his/her progress since the user may access his/her score and see how much he/she has learned from the application through statistical data.
- Ranking:** This feature allows the learner to compare his/her progress with others learners by participating in competitions and seeing their ranking among other users of the application.
- Settings:** This feature allows the user to set the app preferences concerning notifications, sounds and some system options.

Concerning the *Activities* feature, *ProjectEdu* provides theoretical content of Software Project Management and also activities and practices. As Fig. 2 shows, the main topics are:

- Introductory concepts of Project Management;
- Project Management Foundations;
- Project Management Knowledge Areas; and
- Business Environments in Projects.

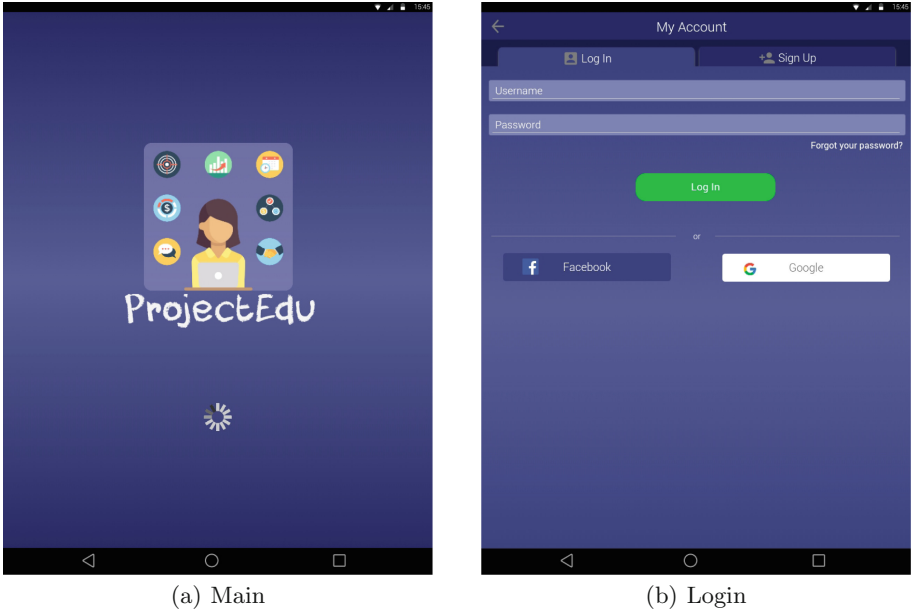


Fig. 1. *ProjectEdu* first screens

Figure 3(a) shows in detail a screen in which the learner is provided with some theoretical content and Fig. 3(b) shows an exercise related to that content.

Regarding *Statistics*, Fig. 4 shows the learner can access some important information about his/her use of the app. For instance, the learner can see how many points were earned, how many days he/she is engaged using the app and the percentage of the content that he/she has already completed. The learner can also follow the daily progress and see how many points he/she earned in each day of the week. The *Ranking*, shown in Fig. 5, shows a global vision of the learners' performance. He/she can see his/her position among all the users.

Although some of these features are usual in mobile learning apps, *ProjectEdu* has been designed considering two artifacts aimed at systematizing the designing of m-learning apps, discussed in Sect. 2. We opted for an iterative and incremental development process, with short phases and proximity to the final target audience, so that the application is well accepted by them. In this sense, the features are inserted and tested gradually.

In the current version of *ProjectEdu*, *ReqML-Catalog* guided the definition of *learning* and *usability* requirements. Since we want to provide a more attractive approach, it is important to consider user interface and usability aspects to achieve this goal.

¹ <https://www.justinmind.com/>.

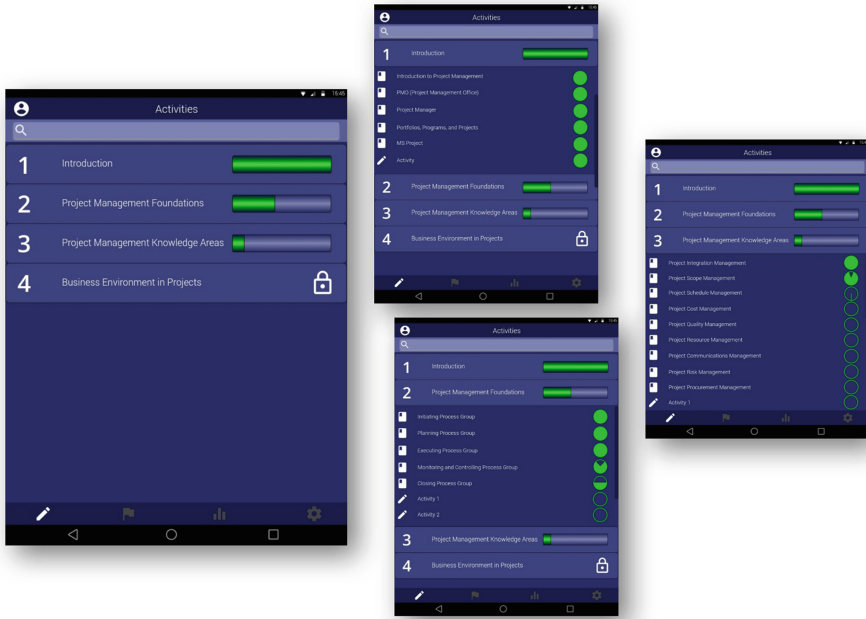


Fig. 2. Topics of project management

ProjectEdu has the following usability requirements suggested by *ReqML-Catalog*: attractiveness, continuity, information presentation, homogeneity of layout and components and concise messages.

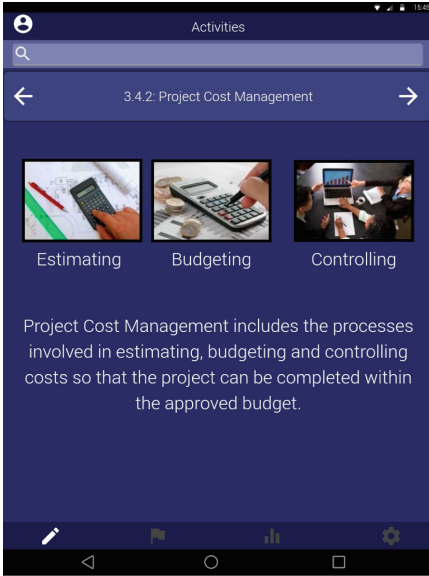
Furthermore, we considered the learning requirements suggested by *ReqML-Catalog*, such as: learning style, knowledge at the right time, educational activities, motivation, engagement and progress tracking.

Progress tracking can be seen applied in the *Activities* feature, which shows a progress bar for each topic, and also in the *Statistics* screen, in which the learner can see his/her progress.

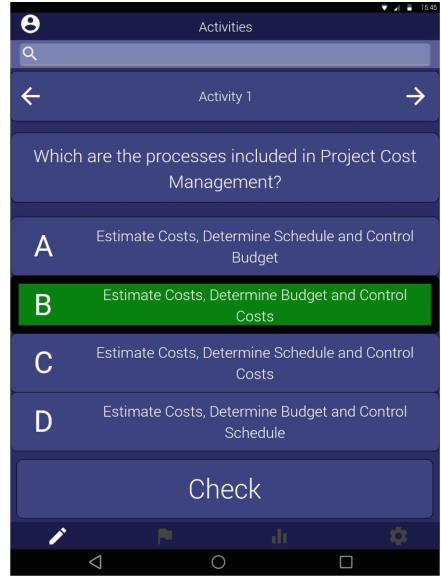
Dealing with motivation, engagement, learning styles and so forth is not an easy task. *MLearning-PL* guided this process of the design by applying pedagogical patterns. Following, we present in Table 1 how each pattern was applied in *ProjectEdu*.

4 Evaluation

Aiming to answer our research question, we chose to carry out the evaluation of *ProjectEdu* conducting a usability test. Usability is most often defined as the ease of use and acceptability of a system for a particular class of users carrying out specific tasks in a specific environment. Ease of use affects the users' performance and their satisfaction, while acceptability affects whether the product is used [8].



(a) Theoretical



(b) Exercise

Fig. 3. *ProjectEdu* activities

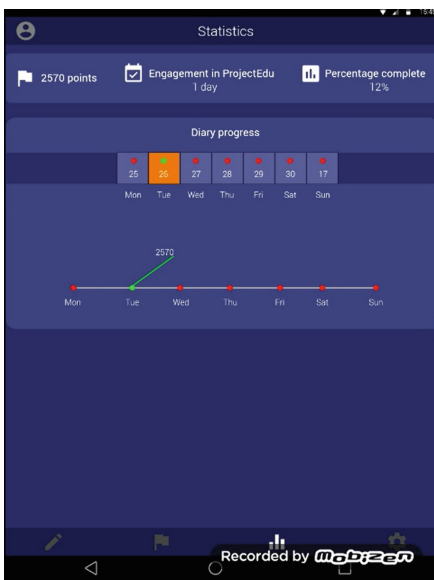


Fig. 4. Statistics

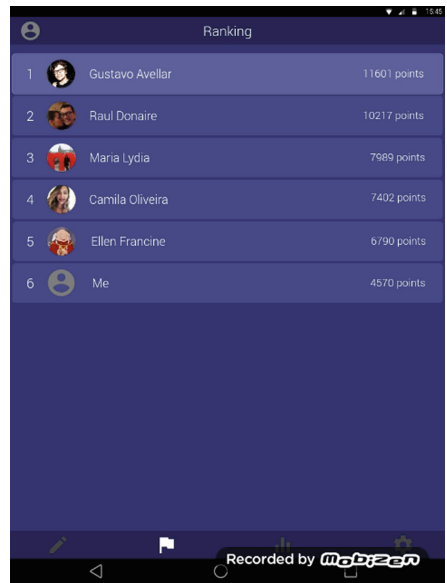


Fig. 5. Ranking

Table 1. Application of each pattern of *MLearning-PL* in *ProjectEdu*

Pattern	Application in <i>ProjectEdu</i>
Be Active	Providing theoretical and hands-on activities
Give Them a Treat	Giving the learners a score when a topic is finished
Gold, Silver and Bronze Medal	Showing publicly the results in the Ranking
Little by Little	Organizing the topics in the smallest unit possible
As Soon As Possible	Providing most important knowledge first
Suitable for You	Providing different types of media, such as images, videos and texts

To ensure a software project has these essential usability characteristics, we used methods we divide into test methods (with end users) and inspection methods (without end users).

4.1 User Tests

Testing with end users is the most fundamental usability method and is in some sense indispensable. It provides direct information about how people use our systems and their exact problems with a specific interface.

We conducted the test with 14 participants throughout an afternoon and early evening in a prepared room from one of our research labs building in the Institute of Mathematics and Computer Science (ICMC), University of São Paulo (USP). During the tests, there were only the researchers and the participant inside the room, moreover we video recorded the user's hands and the tablet screen for further analysis. The participants were undergraduate and graduate students of the Computer Science area from ICMC/USP.

Aiming to characterize the 14 participants of our user tests, we asked them some questions. Participants were firstly asked which type of mobile devices they had: smartphone and/or tablet. As Fig. 6 shows, all of them have smartphones and only three have tablets.

Next, we wanted to know if they have ever used a mobile learning application and as shown in Fig. 7, 79% (11) of the participants have previously used a mobile learning application.

From this participants with previous experience with m-learning apps, we wanted to know how their experience was. Figure 8 shows that 27% had an excellent experience; 55%, i.e, more than half of the experienced participants, had an average experience; 9% had a good experience; and the remainder 9% had a fair experience.

After answering these characterization questions, the user test was divided in three parts: (i) Thinking Aloud; (ii) System Usability Scale; (iii) Open Questions.

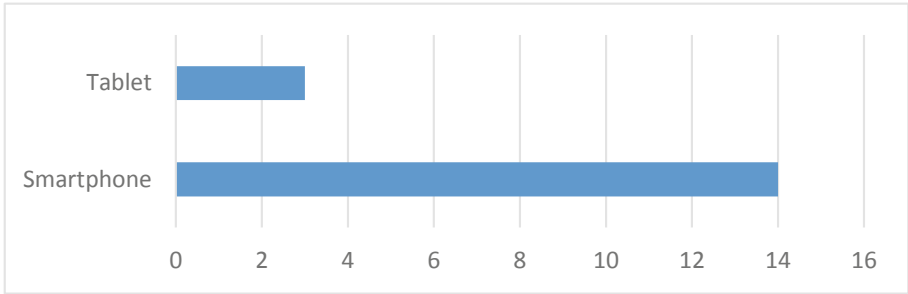


Fig. 6. Which of these mobile devices do you have?

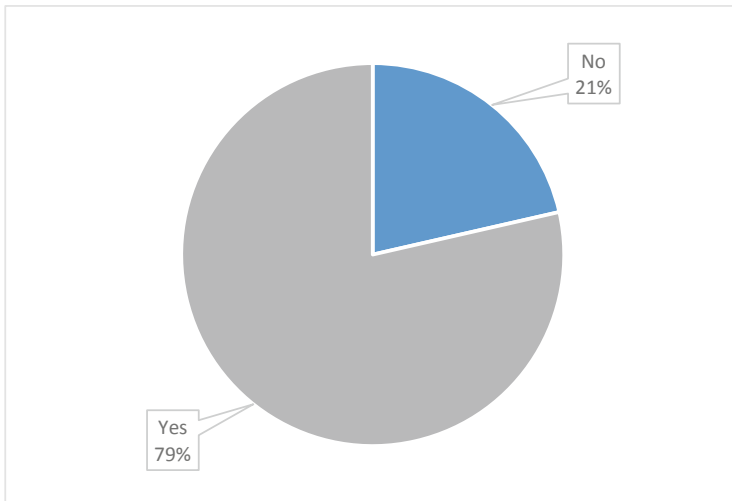


Fig. 7. Have you ever used mobile learning applications?

Thinking Aloud. Thinking aloud (TA) [18] may be the single most valuable usability engineering method. It involves having an end user continuously thinking out loud while using the system. By verbalizing their thoughts, the test users enable us to understand how they view the system, which makes it easier to identify the end users' major misconceptions. By showing how users interpret each individual interface item, TA facilitates a direct understanding of which parts of the dialogue cause the most problems. In TA the time is very important, since the contents of the users' working memory contents are desired.

During this part of the test, the participants followed a set of steps to guide their interaction, available at <https://goo.gl/WJcmJq>.

We based the analysis of our acquired results on some procedures of Grounded Theory [25] to analyze users' comments based on the concept of coding, such as: *open coding* makes possible identification of concepts that are separated into

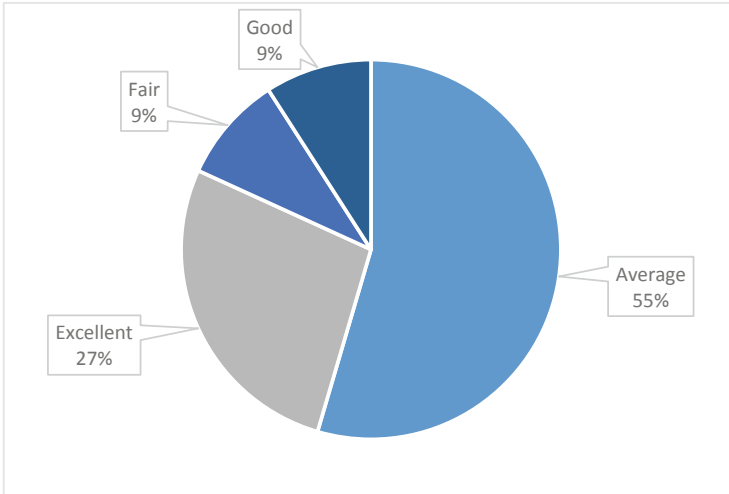


Fig. 8. How was your experience using mobile learning applications?

discrete parts for analysis; and *axial coding* handles connections among codes and groups them according to their similarities.

Our data sample consisted of transcribed recordings from users' TA session. In general, the idea is to provide the users an experience with an m-learning application, in this case *ProjectEdu*. The users were encouraged to constantly verbalize their thoughts and share their opinions on the app functionalities.

All reports were organized on a single file and each sentence was analyzed to derive the codes by using open coding procedures.

Learners controlling the study was one of the main extracted codes. It reinforces the idea that the students using m-learning applications have a need of constant controlling their learning process, task that were previously assigned to instructors in the traditional learning [22]. It is exemplified in the sentences below: *"I think it needs to be very clear when the questions are about to appear. The app should have the option of skipping it if the user is not willing to answer the questions in that moment. Sometimes he/she just want to refresh his/her mind and avoid answering stuff"* and *"I don't know how it would work in a video, but it would be interesting to mark what I have already seen, where I stopped, also that I could comment, in private or public, do notes. I would help my learning process"*

In addition, users described their experience with the feedback of the system: *"I'm not so sure if I finished the last topic. I need to move forward until I get to the end? Now I'm not sure, I was at the question screen, now I don't know if this new screen belongs to the new content"* and *"It would also be nice if I had a sense of how long it takes to finish this, keep clicking 'Next' without knowing when it will stop it's demotivating"*. According to Nielsen [18], this topic is critical to a systems usability, hence such sentences were coded in **System feedback needs to be improved**.

Users also reported their experience with the navigability of the application: “An exit icon is missing”, “I didn’t see if there is a button to return into the activities. It’s not cool to click again all this way through to come back to where I was”, “The navigability of the content is the most misleading part” and “It’s odd that you have to click in the icon again to return. I can get used to that, I guess.”. These information was clustered into **Navigability did not please**.

Another code retrieved was **Red/Yellow/Green have special meaning**. Through the test, users were often expecting that items with red/yellow/green elements had an extra meaning, the concepts of right (green) and wrong (red) were attached to that. This code can be exemplified by the following quotes: “Probably green means that I got right”, “This diagram is presented with borders on different colors, I can’t understand the relation between the border and the content. It’s some kind of priority scheme? A traffic light?” and “Hmm ok! I’ve gotten something here and the statistics and the ranking turned to yellow. Why is it yellow?”.

In summary, we identified eight codes and using axial coding procedures, we aggregated each code into categories based on their similarity. We performed open and axial coding several times aiming to refine the emerging codes and categories. Furthermore, we mitigated an eventual bias in the coding process by discussing the codes and categories among the researchers until they came to an agreement for all the concepts found. The categories are presented next, followed by the assigned codes.

Usability: System feedback needs to be improved, Navigability did not please, Non-Intuitive icons, Red/Yellow/Green have special meaning.

Requirements: Statistics needs clarity and dynamism, Application to be practical/fast/safe, Content must be attractive, Learners controlling study.

The categories and codes emerged through Grounded Theory procedures allowed us to suggest some assumptions from our findings. For example, there is an overall agreement of the users that they enjoy to control how to handle their learning process. They also reported this need of deciding what to study (which topic or content), how to study (reading, videos, exercises) and when to study. And in order to properly establish their routine, they reported a need of two major requirements from the system: feedback and navigability. Feedback provides the real time information that will assist their decisions: *How many topics have I already completed? How many exercises does this activity have? Which questions have I answered correctly?* And navigability is the final piece of this structure, the user needs a fast paced, dynamic and intuitive system in order to fully apply their routine. A clunky and uninformative application can demotivate the student, as could be seen in the quotes.

System Usability Scale. Questionnaires are useful for studying how end users use the system and their preferred features, but need some experience to design. They are an indirect method, since this technique does not study the actual user interface: it only collects the opinions of the users about the interface.

There are numerous surveys available to usability practitioners to aid them in assessing the usability of a product or service. Many of these surveys are used to evaluate specific types of interfaces, while others can be used to evaluate a wider range of interface types. The System Usability Scale (SUS) [9] is one of the surveys that can be used to assess the usability of a variety of products or services. There are several characteristics of the SUS that makes its use attractive. First, it is composed of only ten statements, so it is relatively quick and easy for study participants to complete and for administrators to score. Second, it is nonproprietary, so it is cost effective to use and can be scored very quickly, immediately after completion. Third, the SUS is technology agnostic, which means that it can be used by a broad group of usability practitioners to evaluate almost any type of user interface, including Web sites, cell phones, interactive voice response (IVR) systems (both touch-tone and speech), TV applications, and more. Lastly, the result of the survey is a single score, ranging from 0 to 100, and is relatively easy to understand by a wide range of people from other disciplines who work on project teams.

According to Bangor et al. [7] the average study mean is about 70. Considering that the result for *ProjectEdu* was 75, it was above average. However, we wanted to understand what were the points that brought this score down. Aiming to verify the specific objectives proposed for this research, we used the relation between the quality components indicated by Nielsen and the SUS questions. The results are shown in Fig. 9 and discussed next.

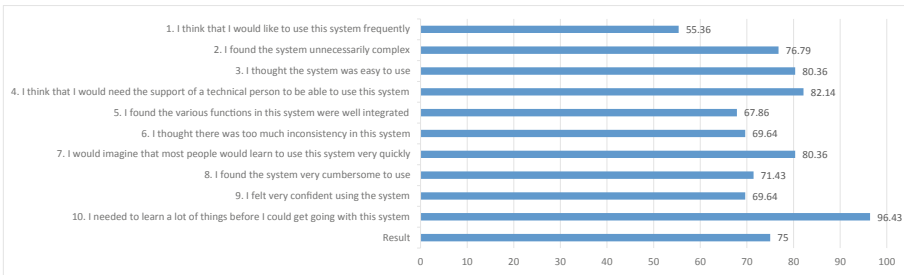


Fig. 9. SUS results

Learnability: learnability is represented in the questions 3, 4, 7 and 10 of SUS. The average of the result of these questions is 84.82, so we can conclude that the users had an easy time learning to use the system.

Efficiency: the items 5, 6 and 8 are related to system efficiency. Analyzing the average of these questions, we obtained 69.64, which means the users consider the system efficient, although the result is slightly inferior to 70.

Memorability: the ease of memorization is assessed by question 2, the score of 76.79 shows satisfaction concerning this item.

Errors: inconsistencies or minimization of errors are measured through question 6. In this item, the SUS score was 69.64, again slightly inferior to 70, but still a relevant result.

Satisfaction: user satisfaction is represented by items 1, 4 and 9. The average of these questions was 69.05, also slightly inferior to 70, but expected since the participants raised some points of improvement.

Overall, the *ProjectEdu* SUS Score of 75 demonstrates that the system meets usability requirements and the quality component analysis gives us indications of the improvement points that should be prioritized, such as system feedback and content navigation.

Personal Opinions. The last questions of the user test took participants' personal opinions about the experience with *ProjectEdu* and also about their mobile learning applications, in general.

First, we asked them to describe if they have faced any difficulties during the use of *ProjectEdu*. Most of them mentioned not facing major difficulties, but some minor difficulties were raised, such as: (i) Statistics screen; (ii) Next button; and (iii) Content navigation.

Next, we asked if they could change *ProjectEdu*, what kind of changes they would make. In addition to the improvements to the items that caused difficulties in the user experience, other interesting improvements were suggested. We can highlight: (i) social network integration; and (ii) a space for adding personal notes.

Proceeding to their experiences with mobile learning applications, in general, we asked, if they would use a mobile learning application to learn a new content in a daily basis. Figure 10 shows that 93% (13) would use and only 7% (1) would not.

When asked the reason why using or not a mobile learning applications, the participants who answered positively mentioned that (i) they already use another m-learning app; (ii) it is easy to use anytime and anywhere; (iii) it is practical and flexible way of learning a new content; and so forth. Regarding the participant who would not engage in a mobile learning app, the reason is not being able to commit to a long-term course.

The last question of the survey took free-text answers about respondents experiences. In general, they reported a pleasant and interactive experience and mentioned *ProjectEdu* is an interesting app that they would definitely use.

4.2 Heuristic Evaluation

Heuristic evaluation (HE) is the most common informal method. It involves having usability specialists judge whether each dialogue or other interactive element follows established usability principles [19].

The original and adopted approach is for each individual evaluator to inspect the interface alone. Only after all the evaluations have been completed are the evaluators allowed to communicate and aggregate their findings. This restriction is important in order to ensure independent and unbiased evaluations. During a single evaluation session, the evaluator goes through the interface several times,

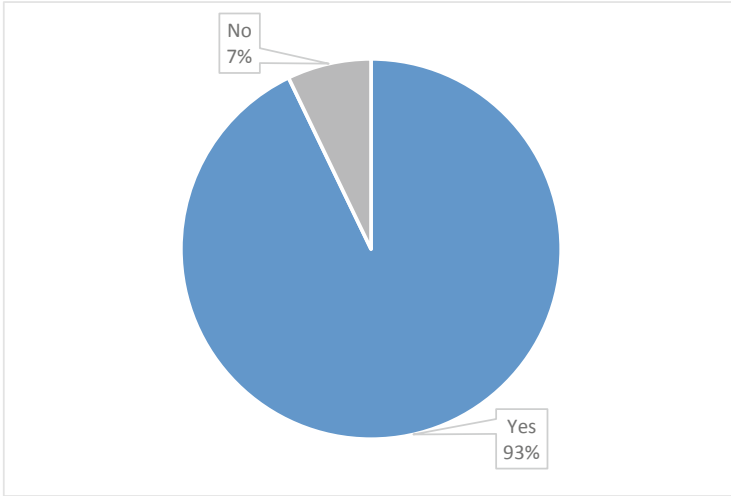


Fig. 10. In your day-to-day life, would you use a mobile learning application to learn a new content?

inspects the various interactive elements, and compares them with a list of recognized usability principles (in this case, Nielsen’s Usability Heuristics [18]).

Our heuristic evaluation was performed by four usability specialists who followed the instructions available at <https://goo.gl/B5FoN9> and fill in a table with the following information:

- ID: Sequential numbering that identifies the problem pointed out by the expert.
- Heuristic: Represents the numbering of each of Nielsen’s heuristics.
 1. Visibility of system status
 2. Match between system and the real world
 3. User control and freedom
 4. Consistency and standards
 5. Error prevention
 6. Recognition rather than recall
 7. Flexibility and efficiency of use
 8. Aesthetic and minimalist design
 9. Help users recognize, diagnose, and recover from errors
 10. Help and documentation
- Description of the problem: Description presented by the expert for the problem found.
- Task: Represents the tasks previously presented.
- Screen: Name that best represents the system screen where the problem was identified.
- Degree of severity:

- 0 = I don't agree that this is a usability problem at all
- 1 = Cosmetic problem only: need not be fixed unless extra time is available on project
- 2 = Minor usability problem: fixing this should be given low priority
- 3 = Major usability problem: important to fix, so should be given high priority
- 4 = Usability catastrophe: imperative to fix this before product can be released

After they completed the individual evaluations and then their findings were aggregated, 42 issues were identified. In Table 2, we highlight the issues which also were mentioned by the users during the user tests and identified using Grounded Theory.

We grouped the average and maximum severities by heuristic (Fig. 11) to analyze the strengths and weaknesses identified.

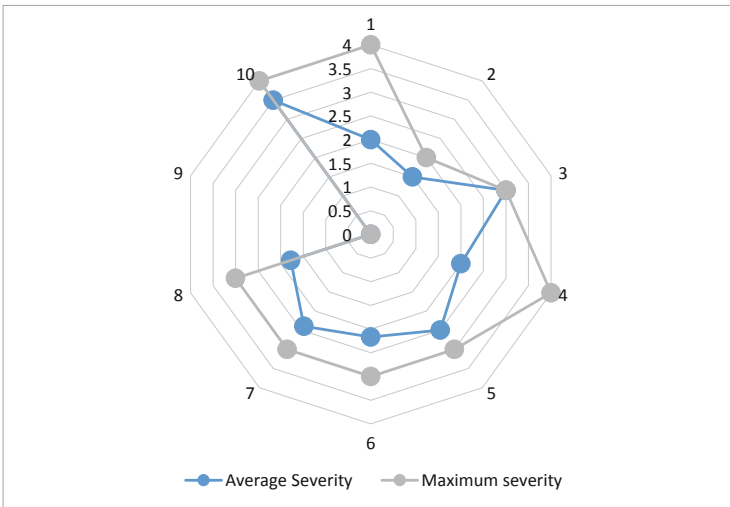


Fig. 11. Nielsen's heuristics vs. severities

As we can see, heuristic #9 regarding *aesthetic and minimalist design* was not a raised concern. On the other hand, three heuristics were violated with maximum severity, which were #1, #4 and #10, regarding, respectively, *visibility of system status*, *consistency and standards*, and *help and documentation*.

Analyzing the comments of the evaluators, we agreed that some improvements must be made in order to evolve *ProjectEdu* and we will definitely take their observations into consideration. On the other hand, most of their suggestions will be solved when *ProjectEdu* will no longer be a prototype, since the issues mentioned are due the prototyping tool.

Table 2. Issues identified in the heuristic evaluation

Heuristic	Description of the problem	Degree of severity
1	Missing sign up confirmation feedback (System feedback needs to be improved)	2
4	The forward arrow does not work (Navigability did not please)	4
2	The icon used to represent the ranking functionality is not intuitive, usually the systems use the icon of a “medal” or award (Non-intuitive icons)	2
4	The “Ranking” and “Statistics” elements of the lower toolbar are in highlighted yellow placement indicating that there have been changes. Badges are usually used above the icon when there are updates or notifications (Red/Yellow/Green have special meaning)	1
6	The first place in the ranking is highlighted by color, if there are many people participating in the ranking it will be difficult to find the classification (Statistics needs clarity and dynamism)	3
6	Explanations of the initial tutorial are displayed one at a time, and if the user clicks in one of the icons in the middle of the tutorial, he/she can not see the explanations again (Application to be practical/fast/safe)	1
6	When there are lots of “Lessons and Activities” in the expansions it is difficult to know which of them is the marker of progress (Content must be attractive)	1
1, 7	Information about the system status is missing. I suggest putting a breadcrumb, since there are sub-activities (Learners controlling study)	3

5 Conclusions and Future Work

This paper has presented an evaluation of a mobile learning application prototype, entitled *ProjectEdu*. In general, users were enthusiastic and positive about the use of mobile learning applications. Although *ProjectEdu* is still a prototype and requires improvements, the evaluated version has fulfilled the requirements of usability.

On the other hand, in order to be as attractive as the users expect and they really feel motivated to use, several improvements still must be made: more attractive content, more dynamic statistics, more intuitive icons, better navigability, feedback, and self-learning mechanisms.

As future work, we aim at considering all the improvement points raised during the evaluations conducted to develop the first version of *ProjectEdu*. Moreover, we intend to conduct other types of evaluations concerning learning aspects while using the mobile app. Shortly, we intend to include other relevant requirements of an m-learning app, using *ReqML-Catalog* as a basis.

Acknowledgements. This study was financed by the University of São Paulo (USP) and the Brazilian funding agencies: Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, CNPq and FAPESP. The authors are also grateful to the HCI experts for their valuable contributions to this work and also to all those who helped by participating in the studies.

References

1. ACM/IEEE-CE: Computer Engineering Curricula (2016). <https://www.acm.org/binaries/content/assets/education/ce2016-final-report.pdf>
2. ACM/IEEE-CS: Curriculum Guidelines for Undergraduate Programs in Computer Science (2013). <https://www.acm.org/binaries/content/assets/education/cs2013-web-final.pdf>
3. ACM/IEEE-IS: Curriculum Guidelines for Undergraduate Degree Programs in Information Systems (2010). <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/is-2010-acm-final.pdf>
4. ACM/IEEE-IT: Curriculum Guidelines for Baccalaureate Degree Programs in Information Technology (2017). <https://www.acm.org/binaries/content/assets/education/curricula-recommendations/it2017.pdf>
5. ACM/IEEE-SE: Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering (2014). <https://www.acm.org/binaries/content/assets/education/se2014.pdf>
6. Alexander, B.: Going nomadic: mobile learning in higher education. *Educause Rev.* **39**(5), 28–35 (2004)
7. Bangor, A., Kortum, P., Miller, J.: Determining what individual SUS scores mean: adding an adjective rating scale. *J. Usability Stud.* **4**(3), 114–123 (2009). <http://dl.acm.org/citation.cfm?id=2835587.2835589>
8. Bevan, N.: Measuring usability as quality of use. *Software Qual. J.* **4**(2), 115–130 (1995)
9. Brooke, J., et al.: SUS-A quick and dirty usability scale. *Usability evaluation in industry* **189**(194), 4–7 (1996)
10. Craig, A., Coldwell-Neilson, J., Goold, A., Beekhuizen, J.: A review of e-learning technologies-opportunities for teaching and learning. In: *CSEDU 2012–4th International Conference on Computer Supported Education*, pp. 29–41. [INSTICC] (2012)
11. Crompton, C.: A historical overview of mobile learning: toward learner-centered education. In: Berge, Z.L., Muilenburg, L.Y. (eds.) *Handbook of Mobile Learning*, pp. 3–14. Routledge, Florence (2013)
12. Farooq, U., Schafer, W., Rosson, M.B., Carroll, J.M.: M-education: bridging the gap of mobile and desktop computing. In: *Proceeding of the IEEE International Workshop on Wireless and Mobile Technologies in Education*, pp. 91–94. IEEE (2002)

13. Fioravanti, M.L., Barbosa, E.F.: A Pedagogical Pattern Language for Mobile Learning Applications. In: Proceedings of 24th Conference on Pattern Languages of Programs (PLOP 2017), October 2017, Vancouver, BC, Canada (2017)
14. Kearney, M., Schuck, S., Burden, K., Aubusson, P.: Viewing mobile learning from a pedagogical perspective. *Res. Learn. Tech.* **20**, 14406 (2012)
15. Keegan, D.: The incorporation of mobile learning into mainstream education and training. In: World Conference on Mobile Learning, p. 11, Cape Town (2005)
16. Kinshuk, S.J., Sutinen, E., Goh, T.: Mobile technologies in support of distance learning. *Asian J. Dist. Educ.* **1**(1), 60–68 (2003)
17. Naismith, L., Lonsdale, P., Vavoula, G.N., Sharples, M.: Mobile technologies and learning (2004)
18. Nielsen, J.: Usability Engineering. Elsevier, New York (1994)
19. Nielsen, J.: Usability inspection methods. In: Conference Companion on Human Factors in Computing Systems, pp. 413–414. ACM, New York (1994)
20. O'Malley, C. et al.: Guidelines for learning/teaching/tutoring in a mobile environment (2005), public deliverable from the MOBILearn project (D.4.1)
21. Quinn, C.: mLearning: Mobile, wireless, in-your-pocket learning. *LiNE Zine* **2006**, 1–2 (2000)
22. Sarrab, M., Alzahrani, A., Alwan, N.A., Alfarraj, O.: From traditional learning into mobile learning in education at the university level: undergraduate students perspective. *Int. J. Mob. Learn. Organ.* **8**(3–4), 167–186 (2014). <https://doi.org/10.1504/IJMLO.2014.067014>
23. Soad, G.W., Fioravanti, M.L., Falvo Júnior, V., Marcolino, A.S., Duarte Filho, N.F., Barbosa, E.F.: ReqML-catalog: the road to a requirements catalog for mobile learning applications. In: Proceedings of the 47th Annual Frontiers in Education Conference (FIE 2017), Indianapolis, Indiana, USA, October 2017
24. Sommerville, I.: Software Engineering, 9th edn. Addison-Wesley Publishing Company, Boston (2010)
25. Strauss, A., Corbin, J.: Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. SAGE Publications, Thousand Oaks (1998). <https://books.google.com.br/books?id=wTwYUnHYsmMC>
26. Svetlana, K., et al.: Adaptation e-learning contents in mobile environment. In: Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, pp. 474–479. ACM (2009)
27. Traxler, J., Leach, J.: Innovative and sustainable mobile learning in Africa. In: Proceedings of the 4th IEEE International Workshop on Wireless, Mobile and Ubiquitous Technology in Education (WMUTE 2006), pp. 98–102. IEEE (2006)
28. UNESCO: Information and communication technology in education: A curriculum for schools and programme of teacher development. UNESCO (2002)
29. Wu, W.H., Wu, Y.C.J., Chen, C.Y., Kao, H.Y., Lin, C.H., Huang, S.H.: Review of trends from mobile learning studies: a meta-analysis. *Comput. Educ.* **59**(2), 817–827 (2012)
30. Zamfirache, V., Olteanu, A., Tapus, N.: Collaborative learning assistant for android. In: 11th Roedunet International Conference (RoEduNet 2013), pp. 1–6. IEEE, Sinaia (2013). <https://doi.org/10.1109/RoEduNet.2013.6511757>