Personal Learning Records: Supporting Metacognition Cycles to Improve the Learning of Programming

Pascal Bos
University of Twente
PO Box 217, 7500 AE Enschede
the Netherlands
p.m.bos@student.utwente.nl

ABSTRACT
This paper describes the development and the investigation of the use of Personal Learning Records (PLR) to support the metacognition cycles. Literature provides evidence that metacognition skills are highly correlated to success in studies, especially for tasks that have a high cognitive load like computer programming and problem solving. PLR aims to support the reflection part of metacognition cycles by means of a structured self-assessment. It does so by providing friendly rubrics to three levels of proficiency (entry, intermediate, and target). The goal is to assist students in improving their metacognitive knowledge, facilitate the development of their metacognition skills and, consequently, improve their performance in a computer programming course. The system was tested empirically using the Communicability Evaluation Method and the results indicate some key parts to be improved.

Keywords
reflection, student, programming, education, personal learning records, PLR, self-assessment, supported learning

1. INTRODUCTION
Perhaps you have studied computer programming yourself or perhaps you know someone who has. Either way, the following phrase is likely familiar to you *"Computer programming is hard!"*. A lot of those who studied computer programming have experienced difficulties [1]–[3]. Therefore, this research aims at improving the metacognitions skills of students of computer programming. This study focuses on the self-assessment aspect of metacognition as self-assessments close the metacognitive cycle, allowing for the consolidation of knowledge and for the planning of future learning.

In the Section 2, we introduce the theoretical foundation for this research. In short, we will approach this problem from the context of Cognitive Learning Theory (CLT) [10], Metacognition [12] and Self-Regulated Learning (SRL) [13]. These theories formed the foundation on which the software was developed.

We develop a high-fidelity functional prototype of the Personal Learning Records to provide students with a structured way of self-assessment. The Research Goals related to this development are presented in Section 3. The tool was designed to be minimalistic, simple, and easy to avoid adding to the extraneous load of the student while adding to their metacognitive understanding and streamlining the self-assessment process. Using the PLR, we try to improve the experience students of computer programming have with self-reflection. Evaluation of this tool will be done according to the Design Science Research (DSR) approach [4], as discussed in Section 4, while the PLR is presented in details in Section 5.

For designing the prototype, an overview of related work will be used to determine sensible design practices, to provide examples of (similar) research and to identify the research gap. Most specifically, it appears that most research of educational uses of metacognition focus on either introducing it to the classroom [5], [6] or on providing support tools for improving metacognition in general. Furthermore, one study was found to compare currently existing support tools for reflection [7]. This overview is also presented in Section 5

The rest of this paper is structured as follows: we present the results are presented in Section 6, Limitations of this research in Section and we discuss the Conclusions and Future Work in Section 8.

28th Twente Student Conference on IT, Febr. 2nd, 2018, Enschede, The Netherlands.

2. THEORETICAL FOUNDATION
Learning computer programming is complex and had been approached in several ways by researchers in computer programming education. In particular, Cognitive Load Theory (CLT) has been used to improve the design of instructional material [8]–[10], especially for introductory courses [11]–[13], while Self-Regulated Learning (SRL) and Metacognition theories have approached the monitoring of the learning process as a way to support it [14]–[16]. The Cognitive Load Theory (CLT) explains that Problem Solving tasks (like computer programming) have a higher cognitive load, and that the germane load varies significantly among students [17], depending on previous experience and on how developed their metacognition skills are – necessary to regulate learning processes that are often lengthy and complex [18], [19]. As part of a bigger research project, the Personal Learning Records aims to support the monitoring of the learning process (the Metacognition). Therefore, this literature review privileges the discussions on Self-Regulated Learning, and specifically, Metacognition.

The SRL model of Boekaerts [20], illustrated in Figure *1*, organizes the 'Regulation of the Learning Process' as an intermediary layer. Metacognition has been used to help regulate the learning process while also offering benefits to the innermost layer by supporting cognitive strategies [18].

Research on SRL and Metacognition has provided evidence that monitoring the cognitive process, which happens in the metacognition cycles, positively affects cognitive processes [15]-[21]. The same effect is sought with the improvement of instructional material following advice from CLT. In that sense, it is worth highlighting the work of Glogger-Frey and colleagues [22] in which they present findings that – after some practice – the self-regulated group presented better results than the group that received direct instruction. Although the guidance provided to the instructed group led to lower extraneous load – which is often sought as a way to improve the quality of the instructional design. It is worth noting that the metacognition approach requires that the students have enough practice time for their results to surpass those that received a well-designed direct instruction. Well-designed instructions bring faster results and tangible benefits like the reduction of the extraneous load. However, the downside of only offering well-designed instructions is that the computer science field requires professionals to learn and adapt quickly to new technologies. Well-designed instructions are often not present to support this need for constant learning and adaptation.

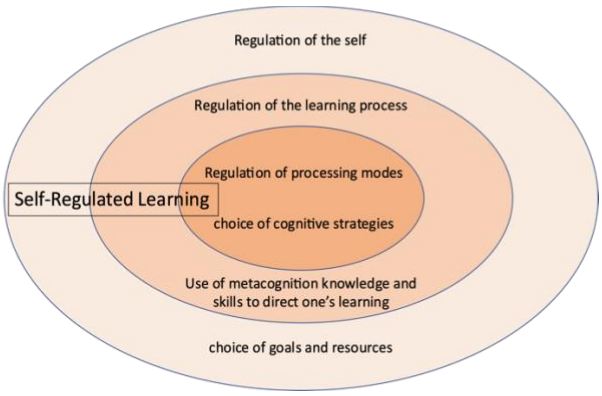


Figure *1. The three-layered model of Self-Regulated Learning (redrawn from the model of Boekaerts in* [20]***).***

Havenga and colleagues argue that a possible reason for the underperformance of students in computer programming may be the fact that *"students mainly focus on the product of programming, namely on computer programs, rather than on the process of programming"* [15] (Page 3). Using Metacognition to improve programming learning seems to be convenient because programming consists of formalizing the process used to solve a problem [21] (Page 2).

3. RESEARCH GOALS

Computer programming is a complex subject to learn. Consequently, it often gets associated with low pass rates and high workloads. The problem becomes even more relevant when new students start deciding against taking a course in computer programming because of it.

Using guided learning processes which incorporate the full metacognition cycle can benefit students greatly in dealing with the demands of learning computer programming. However, it is not trivial to design such processes.

This study investigates the effects of providing a structured self-assessment tool to support such self-assessments for students in computer programming.

Specifically, this research will endeavor to answer the following research question.

RQ? *How does a guided, structured reflection process influence the learning experience of students in computer programming?*

To answer the research question, this study proposes the development of a software tool to provide students in computer programming with a structured means of performing self-assessments as they relate to their studies. This tool offers functionality for students, teachers, and mentors. Mentors are usually closer to students than teachers, have knowledge of the course material and can assist and motivate the students.

This research proposes the following hypothesis.

H1*: The use of PLR will improve the experience students in computer programming have when reflecting on their studies.*

4. EVALUATION

To test our hypothesis, we will follow the Design Science Research (DSR) approach, as described by Peffers (2020) [4], because it is built around the evaluation of artifacts. This makes it ideal for this research as the Personal Learning Records is such an artifact.

The DSR approach follows the following steps.

1. Problem identification & motivation
2. Objectives of a solution
3. Design & development of the artifact
4. Demonstration of the artifact
5. Evaluation of the artifact based on demonstration
6. Communication of the observed results

Figure 2, taken from [4] and adapted for the use-case of this study, gives an overview of this evaluation approach.

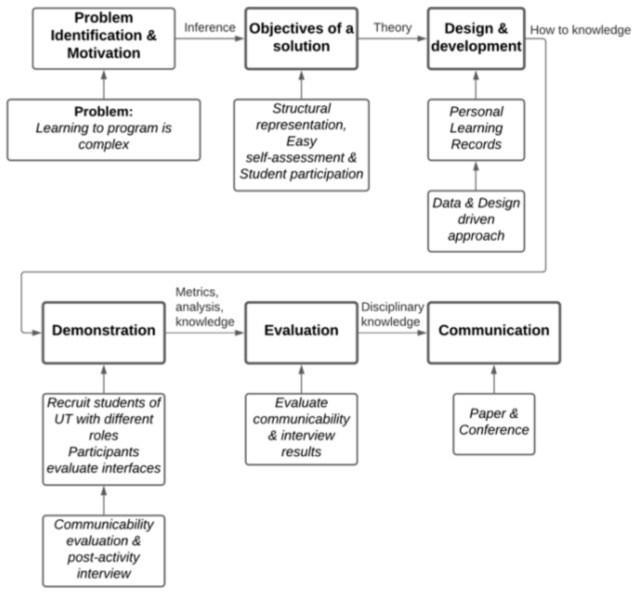


Figure *2. Research approach as an instantiation of DSR as illustrated by* [4]

For evaluating the proposal, the Communicability Evaluation Method [23], [24] was followed. The goal of this procedure is to identify and to be able to explain possible disruptions in communication between the user of a system and the system itself. That is, any point in the interaction between the user and the system where either party does not understand, or does not properly respond to, the input of the other party.

To this end, a series of 'tags' is defined as explained by [23], [24] to identify possible disruptions in communication between the user and the Personal Learning Records as well as to get an understanding of why they happened.

The procedure entails the usage of our proposal in a controlled setting in which a participant will attempt to complete a series of tasks. They are granted some context and a description of the task. The participant is then left to use the system as they see fit with the goal of successfully completing the given tasks. During the activity, participants must follow the "think aloud" technique, that consists of saying what they are thinking and doing, so that we can further analyze the motivations behind the interaction later [25]. Participants must indicate when they believe to have finished the task or if they gave up the task. While working on the tasks, an audio-visual recording of the participant's voice (because of the think aloud technique), the screen and the participant's face will be made. This will later be reviewed during the tagging phase of the communicability method. A consent form is provided to the participants prior to the evaluation.

During this process, the evaluator may be present but should refrain from interacting with the participant when it concerns the task the participant is attempting to complete. Interactions of a personal or otherwise contextual nature can be conducted and are encouraged. The evaluator will take notes on the interaction between the participant and the system. Special focus should be given to observations that may give rise to multiple similar, but distinct, tags. After all tasks have been completed, a post-activity interview will be conducted to obtain any final thoughts from the participant and to clarify any points the evaluator may have noted during the interaction

Once all experiments have been completed, the footage obtained from the participants' screens and faces will be reviewed by the evaluator and the relevant tags will be matched to the periods in the footage where a communication disruption is observed. This footage will then be used as the basis from which to draw the evaluation conclusions regarding whether communication breakdowns were observed and, if so, where, and why they happened. These results will then be used to inform future design.

Regarding the selection procedure of the participants, we wanted our participants to have taken a course in computer programming during which they were asked to perform regular self-assessments. This was done in Module 2 of the Business Information Technology (BIT) Program 2020-2021 and, therefore, participants were randomly selected from that group. Additionally, we recruited participants who were acting as mentors during said course so that we can also evaluate the interface from the point of view of the mentors.

Having set these criteria, we recruited 3 students (Table 1) who had followed a course in computer programming and software design in the academic year 2020-2021 from the University of Twente. These students were asked to perform a self-assessment regarding the topics taught in the course once every two weeks. Furthermore, we recruited 3 students from the University of Twente who had followed this same course in previous year(s) and who were acting as mentors during the course in the academic year 2020-2021.

Table *1. Participant demographic information.*

ID	Age	Gender	Role	Programming Proficiency
1	19	Male	Mentor	3
2	20	Male	Mentee	2
3	25	Male	Mentee	2
4	21	Female	Mentor	2
5	21	Male	Mentor	2
6	20	Male	Mentee	2

Table 1 illustrates the demographic of the selected participants. Programming proficiency was rated by the participants on a scale from -3 (not proficient) to 3 (proficient) [26].

For the sake of clarity, we term the students following the course in computer programming and software design "mentees" and those who were guiding them "mentors". Participants with the Mentor role evaluated both the mentee and the mentor interface while participants with the Mentee role only evaluated the mentee interface.

The tasks we designed for this evaluation are the following.

Table *2. Tasks*

Mentee	Mentor
Find topic 'LT.P.01.01 - Java Syntax Fundamentals' and understand what it is about	Assign syllabus 'Module 2 BIT - Programming Study Unit' to mentee 'Jane'.
Perform a self-assessment for topic 'LT.P.01.01 - Java Syntax Fundamentals'.	Check the progress of mentee 'Jane' on topic 'LT.P.01.01 - Java Syntax Fundamentals'.
Consult the related material of topic 'LT.P.01.02 - Variables and Constants in Java'.	Provide mentee 'Jane' feedback on topic 'LT.P.01.02 - Variables and Constants in Java'.
Look at the feedback provided by your mentor regarding topic 'LT.P.01.03 – Conditionals'.	Post some interesting material for topic 'LT.P.01.02 - Variables and Constants in Java'.

To help the participants understand the context in which they would normally perform such a task and to help them orient their thought-processes. We provided the following scenarios.

Table *3. Contextual prompts.*

Mentee	Mentor
It is the beginning of the week, and you know a topic will be discussed in the next lecture. Before joining, you want some background information.	A new student joined the track late and the teacher has not yet noticed. You decide to save them some work and assign the appropriate syllabus.
A few days have passed since the lecture, and you have practiced with the topic. Now you want to see if you achieved the next level.	You have a mentee who has been struggling or doing very well with a given topic. You are curious about their progress.
You watched a lecture on a topic but are feeling a little lost. You want some extra explanation.	You are done checking the exercises of one of your mentees. They have done very well or missed some things with regards to a given topic and you want to tell them about it.
Your mentor has seen your work and has given you some feedback. You are curious what they said.	You found some interesting material about a given topic and want to share it.

Using these tasks and contextual prompts, we conducted the communicability experiments.

4.1 POST-ACTIVITY INTERVIEW QUESTIONS

This section lists the questions that were used for the post-activity interview. These are later encoded Q1-Q12 in the order they are listed. When discussing the results, these questions are referenced as Q1-Q12. The questions can be found in Box 1.

Q1. Please, assess the mentor-mentee process in M2 BIT for its efforts to teach students programming.

-3 -2 -1 0 1 2 3
Not helpful — Indifferent — Helpful

Q2. Please, assess the degree to which you believe additional software could have benefitted the mentor-mentee process in M2 BIT.

-3 -2 -1 0 1 2 3
Little — Somewhat — A lot

Q3. Please, assess the software support used in the self-assessment system in M2 BIT.

-3 -2 -1 0 1 2 3
Useless — Indifferent — Useful

Q4. Please, assess the frequency of your self-assessments during the programming course of M2 BIT.

-3 -2 -1 0 1 2 3
Seldom — Occasionally — Often

Q5. Please, assess the guidance you received with regards to self-assessment in M2 BIT.

-3 -2 -1 0 1 2 3
Poor — Indifferent — Excellent

Q6. Please, assess the effectivity of the medium through which you performed your self-assessments in M2 BIT.

-3 -2 -1 0 1 2 3
Ineffective — Indifferent — Effective

Q7. Please assess the PLR regarding its use on the self-assessment of a programming course like M2BIT.

-3 -2 -1 0 1 2 3
Useless — Indifferent — Useful

Q8. Please, assess the rate at which you would perform self-assessments if you could use a structured tool, like the Personal Learning Records in a programming course like M2 BIT.

-3 -2 -1 0 1 2 3
Seldom — Occasionally — Often

Q9. Please, assess the difference that using a structured tool for learning, like the Personal Learning Records, would make for you if you could use it in future courses like M2 BIT.

-3 -2 -1 0 1 2 3
Little — Somewhat — A lot

Is there any functionality, currently not present in the Personal Learning Records, that you believe would be essential if it were to be used in a programming course? If so, what would they be?

Is there any functionality, currently present in the Personal Learning Records, that you believe should be presented differently if it were to be used in a programming course? If so, what would they be?

Box 1. Post-activity interview questions

5. PROPOSED SOLUTION: PERSONAL LEARNING RECORDS

The main design screenshots can be found in Appendix A. Here, the design, comparisons with related work, the distribution platform, and security considerations are discussed.

5.1 DESIGN

As mentioned previously, the user-interface (UI) has primarily been designed for mobile devices due to the convenience of a mobile and personal platform and, also, due to some findings from the literature that claims it has benefits for cognitive offloading [27]. Besides that, almost every student is in possession of a mobile device [28] which makes it a natural target. As such, the PLR was designed to accommodate mobile access. In addition, the entire application was designed according to the ten usability heuristics as defined by the Nielsen Norman Group [29].

The landing page (Figure 3) a student sees when they log into the application presents them with the courses they are enrolled in. If there is new feedback available to the student, they will also see a banner indicating that there is new feedback. Clicking on this banner will take them to a feedback page where they can see all the new feedback as well as to which topic it relates.

Clicking on the button saying ‘All Courses’ will take the student to their course overview where they can select a course. Doing so has the same effect as clicking on an ongoing course on the landing page and will take the student to a detail page where they can read the description of the course and see its topics. Additionally, they can see how many topics are at the target level. This widget is aimed to increase students’ awareness and to drive their motivation. Research results have said that progress-awareness is highly related to performance improvement and such a technique has been largely used in wearables (like the Apple Watch) to drive motivation to do exercises[30]. Moreover, each topic is preceded by either an empty star, half star or filled star indicating the current level of the topic. Each star represents, respectively, entry, intermediate and target levels.

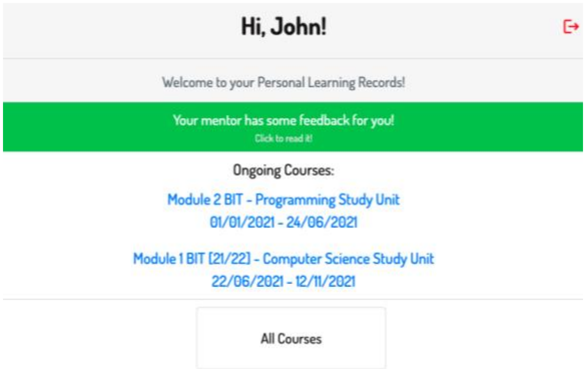


Figure *3. Student Landing Page*

When designing the screens for topics, it was important to look at what information would be needed and what information would be useful to the students. Primarily, the application is designed to help students with their self-reflection. As such, it should be very clear how to get to the screen to perform a self-assessment. For this purpose, two separate links were added. One in the place where iOS users might expect it, namely the app bar. The other here Android users would expect it, a floating action button.

Students might also benefit from a quick reminder as to what the topic is about. For this, three headings were added to include the ‘What?’, ‘Why?’ and ‘When?’ of a topic. That is to say.

- What? What is the topic about and what does it entail?
- Why? Why do we learn about this topic, what is its practical use?
- When? When do we use the knowledge we get from studying this topic?

Research has suggested that clarifying these aspects for students is important when teaching entry-level computer programming [3], [31].

From a topic screen, the student can navigate to the feedback they received about the topic, possible related material, and the self-assessment. As self-assessment is the prime objective of the Personal Learning Records, this screen is also accessible from the feedback and related material pages.

The self-assessment itself (Figure 4) has been kept as simple as possible. Three rubrics are defined to represent the beginner/entry, intermediate and target levels of understanding a topic. Each of these rubrics has a description to indicate what is expected of the student when they have reached that level. This is to allow students to compare what they are capable of to what they should be capable of.

For teachers and mentors, a separate portal has been created (to which they are redirected when logging in) to allow them controlled access to the data produced and needed by the application.

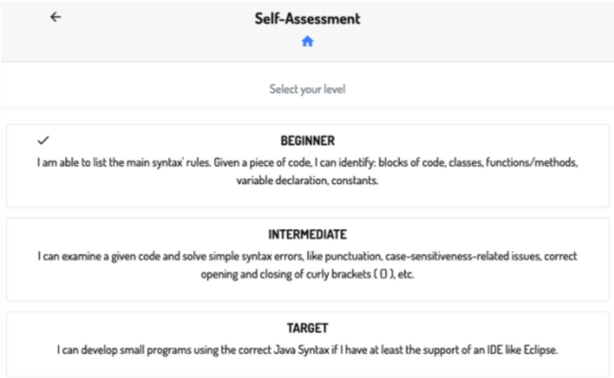


Figure *4. Self-assessment screen for mentees.*

Finally, PLR has been kept as minimalistic as possible to provide students with an environment they can easily navigate. Wherever possible, shortcuts are introduced but only when relevant. No external links to other webpages are present when it can be avoided and, when they are, they will not impact the current state of the application. Lastly, every attempt was made to keep the click paths to a minimum length. These considerations are important to students when interacting with mobile applications [32].

5.2 RELATED WORK

For discussing the related work, a Concept Matrix was constructed with the topics discussed by each paper. This matrix can be found in Figure 5.

Upon examination of Figure 5 and the literature, it follows that the state-of-the-art research on Metacognition includes significant scientific literature produced since the 1960s, as discussed in Section 2. Although Metacognition has become popular among Computer Science educators since the 1980s, it is worth noting that theoretical constructs specific to Computer Programming are still seldomly found in the literature [21], [33]. Considering the support to Metacognition cycles, the available studies do not actively discuss the rate at which students reflect on their studies – the final step of each cycle. This present research aims to increase the understanding of how to best support the Metacognition cycles for Programming students.

Mitrovic et al. [34] investigated the effect of open student models on self-assessment and found positive results. They also posit learning-involvement in decisions made about the student model.

Later, Scheiter et al. [35] would investigate potential bias in mental effort appraisals when viewed from the metacognitive perspective. It appears they only got partial confirmation in this regard, though it may say something for having proper guidance throughout a learning track in which self-assessment is practiced.

In 2009, Bannert et al. [36] investigated the effects of providing a support device to support metacognition in educational environments. While their focus was on metacognitive skills in general, and not on self-assessment, they found positive results.

Regarding software tools to support reflection, Leinonen et al. [7] that describe the Participatory Design process of two mobile apps: TeamUp and ReFlex. Both tools focus on short (60 seconds) audio-visual recordings, used to construct a timeline of reflections for each student or each group, depending on the app. They differ on the fact that ReFlex is for individual use, while TeamUp is used for supporting team reflection. The authors described their Participatory Design experience that included 165 teachers of 13 European countries. The general perception of the participants is that both tools were effective in helping students want to reflect more. However, the study goals were linked to the design of the tools in terms of their features, for instance, whether the limit of 60 seconds per video was ideal and whether the video recording feature would help students develop certain 21st Century Skills. The study didn't report the effectiveness of these tools, in terms of the rate with which the tools affected the frequency and regularity of reflections.

Other studies discuss computer/application-supported reflection in a professional or non-educational environment. Renner et al. (2020) [37] found computer-supported reflection in a work environment to be beneficial for employees' work experience and Renner et al. (2014) [38] found that web-based applications for

	Mitrovic, & Martin (2007). [34]	MacCann-Alfaro et al. (2019). [6]	Renner et al. (2014). [38]	Scheiter et al. (2020). [35]	TeamUp & Reflex from Leinonen et al. (2014). [7]	Bannert et al. (2009). [36]	de Bruin & van Gog (2012). [5]	Renner et al. (2020) [37]
Open Student Models	x							
Metacognition	x	x	x	x		x	x	
Self-assessment	x		x	x	x		x	x
Domain modelling	x							
Positive results	x	x	x	x	x	x	x	x
Argumentation		x						
Quantitative	x	x	x					x
Qualitative		x	x		x	x		x
Software	x	x	x		x	x		x
Educational learning	x	x			x	x	x	
ICT	x	x			x	x		x
Organizational learning			x					x
Job satisfaction			x					
Learner-involvement in decision-making	x		x				x	
Personal goals			x					
Cognitive Load Theory				x				
Reliability of reflection of effort				x				
Self-Regulated Learning				x			x	
Partial results				x		x		
Literature review				x			x	x
Design					x	x		
Social					x			
Guided reflection					x			x
Feedback							x	

Figure *5. Related Work – Concept Matrix*

supporting reflection increased collaborative reflection for hospital staff.

Finally, an Internet search leads to a small number of existing applications that have been developed for supporting reflection and some more that are used for supporting reflection without being built for it. Unfortunately, no literature comparing the differences between the options currently available could be found [39].

5.3 COMPARING TO RELATED WORK

Comparing to the evaluations of TeamUp and Reflex [7], this research focuses more on individual reflection and leaves the use of group-motivation out of scope. In this sense, it has more in common with Reflex, as it also focuses on individual reflection, than it does with TeamUp.

Still, the approaches are distinct. Where TeamUp and Reflex place a focus on audio-visual recordings of reflections in a short timeframe, the Personal Learning Records provides a way of structuring an existing syllabus to allow for reflection on a topic-by-topic basis with clear guidelines of what constitutes a level. This study argues that this may be more convenient for large groups of students as the teacher would not have to listen to all recordings but could get an aggregate overview in a more symmetric manner. For students, it provides structure and quick access to materials related to a given topic.

5.4 DISTRIBUTION

The Personal Learning Records (PLR) has been made into a web application because it is a universally accessible platform in the environment of our research, does not require the user to download and install the application, and the PLR does not require access to the underlying device hardware [40]. Additionally, it is possible to convert the web application into a Progressive Web App (PWA) which can be installed directly on mobile devices and personal computers [41]. Though the web is the preferred platform for this application, it should be noted that the application has been developed with mobile devices in mind.

5.5 SECURITY

Given that the Personal Learning Records was designed to be used in an educational context, significant thought has been given to security. At a minimum, the application will run on a secure environment supporting the HTTPS web protocol.

Seeing that the application has different intended users, security considerations have been made for each of these user categories.

Administrators

The system ships with one predefined administrative user who has full resource access. This user will not have access to the underlying environment, and it is recommended that the password be changed immediately when the PLR is initialized.

Teachers

Teacher accounts can be created by the administrative accounts and have limited resource access.

Mentors

Mentor accounts should be created by teachers but can also be created by administrators. These accounts also have limited resource access. Except for some minor overlap, Teachers and Mentors have access to different types of resources.

Mentees

Mentee accounts can be created by all three of the preceding account types. They have severely limited resource access and can only interact with already existing resources. They cannot delete or create resources.

API Clients

API Clients are the sole user type with access to the API. An API Client account can be created by an Administrator. These accounts can then request an API key from the system which will allow them access to specific sections of the API.

In general, no password is stored as plain-text in the database and API keys are also similarly hashed to avoid sensitive data breaches. Furthermore, user input is validated against validation constraints before reaching the database and, when it does reach the database, built-in drivers are used to ensure that no malicious entries can be inserted into the database.

6. RESULTS

The results from the qualitative evaluation that was performed are divided in.

1. Communicability
2. Interview question analysis.
3. Interview topic analysis.

6.1 COMMUNICABILITY

The results of the communicability evaluation are summarized in figures 6 and 7. Figure 6 denotes the results of the communicability evaluation of the mentee interface while Figure 7 denotes them for the mentor interface.

	Navigation	Meaning assignment	Task accomplishment	Declination/ Missing of affordance	Mentee Interface
I can't do it.			3		
Looks fine to me...					
Help!			3		
Where is? What now?	3				
What's this? Object or action?		3			
Why doesn't it? What happened?		6			
I can't do it this way. Where am I?	6				
Thanks, but no, thanks. I can do otherwise.				2	

Legend: 1-5; >= 5

Figure ***6. Communicability evaluation of mentees' interface***.

Figure 6 maps the predefined tags to predefined HCI ontologies of problems or design guidelines as outlined by [23]. This allows designers to focus on the areas of the design that are the most problematic. The depth of the color indicates the prevalence of the tags as indicated by the legend on the right side.

Navigation. It was observed that incorrect assumptions were observed when navigating towards the self-assessment on the topic detail screen. Sometimes, searching behavior was observed regarding navigating to the self-assessment. This behavior was more common when participants attempted the tasks with only contextual prompts rather than a goal formulation and was, in this case, observed in the first three contextual prompts.

Meaning assignment. A small number of investigative behaviors was observed. Specifically, regarding pictographic elements such as the topic progress stars, the floating action button and the log out button. Of these three, the log out button prompted the least investigation.

Furthermore, a consistent lack of perceived feedback was observed regarding the self-assessment task, contextual or not. Besides this, one participant expressed their need of a screen for the feedback history, instead of the unready ones only.

Task accomplishment. Occasionally, participants indicated that a task could not be completed or requested help with a given task. Noteworthy is that this happened most often during contextual prompts with one occurrence in the self-assessment task.

Declination/missing of affordance. Missed affordances were among the most observed occurrences. This was observed when navigating to syllabi with participants opting for the "All courses" to "Course detail" route instead of making use of the "Ongoing courses" to move to the course detail directly.

	Navigation	Meaning assignment	Task accomplishment	Declination/ Missing of affordance
I can't do it.			1	
Looks fine to me...			3	
Help!			1	
Where is? What now?	6			
What's this? Object or action?		6		
Why doesn't it? What happened?		3		
I can't do it this way. Where am I?	9	1		
Thanks, but no, thanks. I can do otherwise.				2

Mentor Interface: 1-4; 5-8; >=9

Figure *7. Communicability results of mentor interface.*

The mapping of Figure 7 works the same as the one in Figure 6.

Navigation. Disruptions around navigation were the most often observed. Participants demonstrated consistent searching behavior, though some of it was regarding contextual prompts and others regarding topical ones. Participants realized quickly when something would not do what they want but had to realize this often.

Meaning assignment. Ambiguity regarding elements on the interface was observed among all participants but with varying frequency. Often, this related to untruncated text obscuring certain elements or to pictographic elements. Participants also demonstrated frustration when certain interface elements did not lead to the expected result. Sometimes, this feeling was justified. Other times, they had not found the element that would fulfill their expectation. Once, a participant performed the correct series of actions to fulfill a task but did so in the wrong context.

Task accomplishment. Overall, the participants could complete most tasks without help. When they did get stuck, it was often while checking a mentee's progress or while providing feedback. It was more common that a task appeared complete to a participant while additional actions would be needed (happened for only one participant).

Declination/missing of affordance. While not occurring often, it was observed that some participants missed provided affordances as they seemed to be unable to predict the outcome. When affordances were used, they were not declined afterwards.

6.2 INTERVIEW QUESTION ANALYSIS

The post-activity interview included questions for which the participants rated their opinion on a scale, as explained in section 4. A visualization of the distributions of these answers can be found in Figure 8 below.

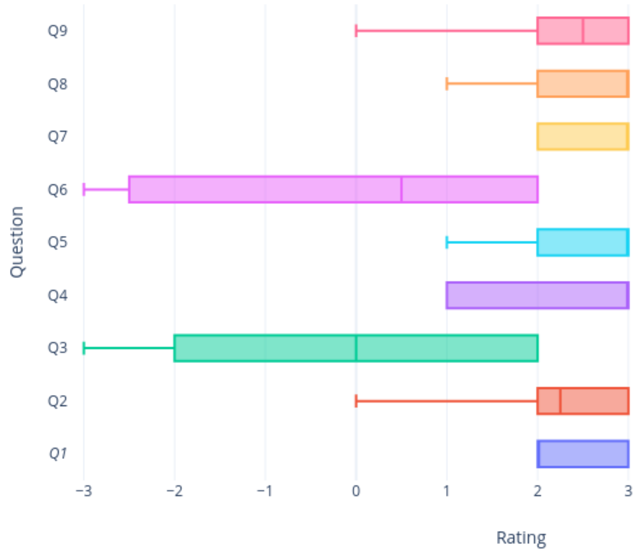


Figure *8. Box plots of the distribution of interview answers.*

The questions can be found on the X-axis and are encoded Q1 through Q9. These represent the questions as set out in section 4.1. The ratings the participants assigned to each question can be found on the Y-axis. Some of the questions directly relate to the experience the participants had with the programming course in Module 2 of the BIT program at the University of Twente [42].

Q1. *Please, assess the mentor-mentee process in M2 BIT for its efforts to teach students programming.* Overall, a positive tendency is observed with regards to the overall helpfulness of the mentoring schema as it was experienced in Module 2 of the BIT program at the University of Twente, with all participants rating it between a two and a three.

This is further reinforced when the more detailed remarks of the post-activity interview are considered. It was quite often mentioned that the previously experienced mentoring schema was helpful to the participant and/or to students they know from the same course. Moreover, one mentor specifically mentioned that the mentoring schema *"made them [students] feel engaged in the [learning] process" (participant 4).*

Furthermore, it was mentioned that mentor interactions helped students overcome their initial fear of the course. The course was known for its difficulty. One participant mentioned that *"a lot of the students get told immediately that module is quite hard" (participant 2).* The "checkpoint" meetings that were organized on a biweekly basis to discuss students' self-assessment were, for some, used as a learning guide.

Lastly, one noteworthy observation made by a participant depicted the mentoring schema not as a teaching instrument but as a tool that draws students closer to learning. Specifically, they mentioned that the mentoring schema *"is a lot better for actually making […] students feel like really drawn to programming so they can learn it on their own" (participant 2).*

Q2. *Please, assess the degree to which you believe additional software could have benefitted the mentor-mentee process in M2 BIT.*
Regarding the benefit of additional software, we observe a neutral

to positive tendency. On the neutral side, it was mentioned by a participant that one should be cautious of introducing additional software into the course as, currently, they felt that there were too many systems that students needed to use already. They mentioned that *"je wordt al doodgegooid met evaluatiesystemen (our translation – you are already being barraged with evaluation systems) (participant 3).* On the positive side, thoughts were dominated by a feeling that the current software supporting the mentoring schema was inadequate with participants mentioning that *"the Form really just felt like a checklist" and that it was "just a separate thing" (participant 1 & 3).*

Q3. *Please, assess the software support used in the self-assessment system in M2 BIT.* Opinions regarding the software supporting the self-assessment system that was experienced previously by the participants are mixed. Most participants agree that it got the job done. Examples of this are that it allowed students to keep track of their progress and understand what they need to do/know for a given topic. Participants were quoted saying *"honestly, it was effective"* and that *"it was genuinely interesting to see what the questions are" (participant 1 & 2).*

Nevertheless, some grievances were expressed as well. Examples of these include that the frequency of self-assessments should depend on study load instead of being constant, 'checkpoint' meetings with the mentors could be more frequent but shorter and that the software used to perform the assessment was not user-friendly. Participants said that *"the system definitely could have benefitted […] being more user-friendly" (participant 1), "sometimes it was a bit slow to work" (participant 5)* and *"I don't think it's [the self-assessment frequency and course load] proportional" (participant 1).*

Furthermore, participants indicated that they felt the self-assessments were disconnected from the rest of the course by having a separate application purely for self-assessment. They also felt that sending links to provide access to the self-assessments is not ideal as, according to them, *"a lot of students don't check their emails" (participant 1)*. A mentor also indicated that it became *"hard for us [mentors] […] to see what everybody was doing" (participant 4)* due to faulty interactions between the systems used to process self-assessments.

Q4. *Please, assess the frequency of your self-assessments during the programming course of M2 BIT.* The participants indicated that they performed self-assessments regularly but with varied frequency. On the most positive end, a participant was quoted as saying: *"I found myself each week updating it [self-assessment], just so that I can see if it's kept up to pace" (participant 2).* In the course they were performing self-assessments, these assessments occurred once a week. First, performing the self-assessment is positive. A common reason for performing the self-assessments was to check progress. Similarly, another participant was quoted as saying *"I wasn't using it for personal development" (participant 6)*, but they were using it, nonetheless.

Reasons for not performing self-assessments as frequently were that the course load dictated the 'relevance' of the self-assessment in the eyes of some participants. Besides course load, extra-curricular activities such as jobs were mentioned as contributing to reduce the value being placed on the assessments. Participants said that they *"wouldn't perform it that well […] just because I have two jobs and the studies, so my schedule is pretty busy"* (participant 1).

Q5. *Please, assess the guidance you received with regards to self-assessment in M2 BIT.* All participants seem to lean toward to positive side when looking back at the guidance they received regarding self-assessments. Some participants said that *"all mentors are very supportive" and "I could always reach my mentor"* (participants 1 & 6). They also indicated that they appreciated always having someone to turn to who could provide undivided attention. Mentors felt that the self-assessment system was sufficiently introduced, saying *"the guidance I received, it was good" (participant 5).*

Q6. *Please, assess the effectivity of the medium through which you performed your self-assessments in M2 BIT.* Opinions on the effectivity of the previously experienced medium used for self-assessment are mixed. On the positive end, participants indicated that it was effective as it got the job done, saying that *"it was effective" (participant 1).* On the lower end, participants indicated that, for them, the lack of efficiency from the software they used stripped it of its effectivity with one participant saying *"so Google Forms, ineffective […], I mean sometime's fine, but it's just not a good way to do it"* after mentioning *"in general, it wasn't something that made me feel like […] it's just this thing I have to do" (participant 2).* The participants who responded positively also shared the sentiment that the software was not efficient, saying *"efficiënt is een ander verhaal" (our translation - efficient is a different story, participant 3).*

Common complaints were that it was slow, *"sometimes it was a bit slow to work" (participant 5)*, that participants had to search before they could make the desired change, *"I'd have to look […] see it's a different student, close it back (participant 1)"*, that the mentees of one participating mentor made their own alternative, *"they copied all of the learning objectives into their own Excel and shared this with me" (participant 1)*, that it felt like a checklist to fill out, and that participants were careful to double-check their input before submitting it, fearing sending a wrong assessment. A mentor indicated on behalf of their mentees, *"let's try not to send by mistake" (participant 5)*. Rarer complaints included mentors *"having to, plenty of times, go to the webmaster" (participant 1)* to get problems with the system resolved.

Q7. *Please assess the PLR regarding its use on the self-assessment of a programming course like M2BIT.* As with Q1, a positive tendency is observed when participants were asked about the usefulness of the PLR for self-assessments in a course like the one they experienced previously. The most common remarks were within the context of usability as indicated by the ease-of-use tag illustrated in Figure 10. Some of the participants were quoted as saying: *"I could find information faster" (participant 4), "everything is structured really well" (participant 2)* or *"so easy to change stuff in" (participant 1)*; the remarks regarding the structure related to the mentee interface.

Q8. *Please, assess the rate at which you would perform self-assessments if you could use a structured tool, like the Personal Learning Records in a programming course like M2 BIT.* The results of Q8 seem comparable to Q4 at first glance, but it would be remiss not to point out that the concentration of answers seems to accumulate around scores 2-3 in Q8, whereas, in Q4, they appear more evenly distributed along the scores 1-3. This is visible in Figure 8 when looking closely at Q8 and Q4. When compared directly, it appears that participants would assess themselves about the same as they already did, saying *"I would use it when it is necessary"* or *"I would probably still do it really often" (participants 5 and 4).*

Q9. *Please, assess the difference that using a structured tool for learning, like the Personal Learning Records, would make for you if you could use it in future courses like M2 BIT.*

The responses from the participants appear to follow a mostly positive tendency, ranging from 0-3 with the highest concentration around 2-3. Most participants agree that a software comparable to the proposed solution would make a difference for them. One of the areas where participants indicated such an application would make a difference is motivation as one mentee indicated that *"if they [students] see they even have a software for them to help them develop, I think they're even going to be more motivated" (participant 6).*

Some participants indicated that the structured format and better relation of the self-assessments to the course information could increase their understanding of the course and provide a helpful overview. One participant mentioned that *"everybody could have a better understanding and […] have a clear overview" (participant 4).*

Q10. *Is there any functionality, currently not present in the Personal Learning Records, that you believe would be essential if it were to be used in a programming course? If so, what would they be?*
When prompted with whether the participants believed some essential functionality is missing from the proposed solution for it to be used in a course like the one they experienced, some remarks were made.

One suggestion was that mentors should be able to assess mentees' understanding of a given topic as they could do so in the system they used previously*, "where do I think the students are at?" (participant 1)*. Another suggestion was to provide a chat-like function for mentees to quickly contact their mentor, *"functionality that can be added […] would be a chat" (participant 5, indirectly participant 6).* Furthermore, several participants mentioned they would want to see an overview which indicates their overall standing in a course, *"een overzichtje van waar je nou eigenlijk in het geheel staat" (our translation – an overview indicating where you are in the whole, participant 3 – indirectly participants 4 & 1).* Lastly, a participant mentioned that a pop-up when saving self-assessments is desirable *"doe dan een pop-up met saved" (our translation – then throw a pop-up saying 'saved', participant 3).*

Q11. *Is there any functionality, currently present in the Personal Learning Records, that you believe should be presented differently if it were to be used in a programming course? If so, what would they be?*
There are several improvements that were suggested by the participants. Most of them surround the mentor interface and relate to the routing through the interface. Specifically, said routing appeared to confuse them occasionally, with one participant saying, *"the succession of actions seems to be very specific" (participant 1)*. Another suggestion was to make use of the dashboard that is present in the mentor interface, *"I think the dashboard is a gold mine" (participant 1)*. A participant also suggested that text could be truncated better. Indicating that *"the only thing that was strange, was the scrolling [to the side] to find the mentee" (participant 4).*

Regarding the mentee interface, the most common suggestions were that the feedback and self-assessment functionalities should be differentiated better and that the number of buttons leading to self-assessment should be reduced. *"I was confused of the fact that the […] assessment and the rubric was on top, and the other button was down. […] It should be somewhere along the lines of related material and feedback" (participant 4)*. *"The part where you […] give your own assessment and the part where you receive your feedback. I confused for a second" (participant 2)*. Another suggestion was maintaining a static rubric ordering, *"the beginner, intermediate and target, keep them just the same [position]" (participant 4).*

6.3 INTERVIEW TOPIC ANALYSIS

Figure ***9. Post-activity interview word cloud.***

Figure 9 indicates the frequency with which certain relevant terms occurred during the post-activity interviews. The larger the term, the more frequently it came up.

Participants were mostly talking about their study program (BIT, Business Information Technology) and how it relates to the self-assessments, mentoring schema, and the PLR. The terms "difference", "personal learning record", "structured tools" and "new system" all relate to their experience with the PLR. Regarding the Google Forms used for self-assessment in the participants' previous experience, the terms "many problem", "checklist" and "email" also relate to this software.

Figure ***10. Post-activity tag cloud.***

The figure above works according to the same principle as Figure 9, except that it demonstrates the prevalence of the tags observed in the post-activity interviews. Here "o.s." stands for "old system" or the system that the participants experienced previously. The ease-of-use of the PLR was among the most discussed tags. Everything else besides "before o.s." appears to be discussed quite evenly overall.

7. LIMITATIONS

This study was meant to explore an area of potential benefits for software tools like the PLR to support learning and reflection. As such, there are some limitations that should be mentioned.

Given the sample size of the participants (N = 6), this study did not reach the saturation point for qualitative analysis. However, this only means that this study may not have caught all possible insights. As such, this research seeks to provide a basis for future work. While the results of this study may provide some insights, they should still be validated in future work.

Finally, due to the timespan of this study, the Personal Learning Records was developed in a span of two weeks and is currently only suitable for small-scale controlled testing.

8. CONCLUSIONS & FUTURE WORK

This study set out to answer the following research question.

To what extent does a guided reflection process, supported by a software tool for structured reflection, influence the experience of reflecting for students in computer programming?

The presumed hypothesis was as follows.

The use of PLR will improve the experience students in computer programming have when reflecting on their studies.

The results obtained through the communicability analysis and the post-activity interview appear to support the hypothesis given that the concentrations of reported answers shifted to the upper spectrum when related to the PLR instead of the previously experienced system.

Furthermore, through the analysis of the interviews, it appears that the participants are more favorably disposed to the proposed system when compared to their prior experience.

As such, the conclusion is that a tool like the PLR may improve the experience of students and is worth pursuing further, based on the results obtained through this research.

Zooming in on the PLR, it should be noted that the participants evaluating the mentee interface, and, to some extent, those evaluating the mentor interface indicated that they appreciated the structure the tool provided. Moreover, this was also the most common compliment for the system they used previously. While it may be premature to conclude that a well-structured tool linking course material to assessments is valued by students, it could be argued that there are signs pointing in this direction.

However, based on the consistency of navigational disruptions, it appears that there are some structural weaknesses in the current design of the PLR. The mentor interface seems to need some consideration based on the locations of these navigational disruptions. The mentee interface appears to be structured well but some thought should be given to the placement of certain elements, such as the self-assessment buttons. Specifically, these disruptions may jeopardize the aim of providing easy access to the self-assessment screens.

Turning to the potential for future works, based on the results from this evaluation, it would be interesting to extend it to a larger-scale evaluation with significantly more participants using the proposed system for a longer period. To this end, the PLR or a similar tool should be further developed beforehand given that it does not currently support such large-scale usage.

Moreover, an Application Programming Interface (API) should be made available for the PLR as well to allow for the possibility of integration with other applications and for easy extension of the PLR itself.

Additionally, it would be interesting to see how students feel about the multitude of software they use for their educational activities. The reasoning behind this is that, in this study, there was a participant cautioning the introduction of additional software. Given that this research was operating on a small scale, it may well be that more students feel similarly; making this an interesting avenue of research to pursue when considering distribution and integration.

Finally, it may be interesting to find out the best ways to integrate software such as the PLR into the educational context. This study piloted at a Dutch university, but educational contexts can vary even in the same country. As such, it would be valuable to know how different educational contexts look at developments such as the PLR.

9. REFERENCES

[1] T. Jenkins, 'ON THE DIFFICULTY OF LEARNING TO PROGRAM', 2002. doi: 10.1.1.596.9994.

[2] Y. Bosse and M. A. Gerosa, 'Why is programming so difficult to learn?', *ACM SIGSOFT Softw. Eng. Notes*, vol. 41, no. 6, pp. 1–6, Jan. 2017, doi: 10.1145/3011286.3011301.

[3] C. S. Cheah, 'Factors Contributing to the Difficulties in Teaching and Learning of Computer Programming: A Literature Review', *Contemp. Educ. Technol.*, vol. 12, no. 2, May 2020, doi: 10.30935/cedtech/8247.

[4] K. Peffers *et al.*, 'THE DESIGN SCIENCE RESEARCH PROCESS: A MODEL FOR PRODUCING AND PRESENTING INFORMATION SYSTEMS RESEARCH', Jun. 2020. Accessed: Apr. 26, 2021. [Online]. Available: https://arxiv.org/abs/2006.02763.

[5] A. B. H. de Bruin and T. van Gog, 'Improving self-monitoring and self-regulation: From cognitive psychology to the classroom', *Learning and Instruction*, vol. 22, no. 4. pp. 245–252, Aug. 2012, doi: 10.1016/j.learninstruc.2012.01.003.

[6] N. MacCann-Alfaro, A. García-Valcárcel, and E. M. Morales-Morga, 'Use of ICT to promote and analyze argumentative and metacognitive processes in higher education', in *ACM International Conference Proceeding Series*, Oct. 2019, pp. 1028–1033, doi: 10.1145/3362789.3362927.

[7] T. Leinonen, A. Keune, M. Veermans, and T. Toikkanen, 'Mobile apps for reflection in learning: A design research in K-12 education', *Br. J. Educ. Technol.*, vol. 47, no. 1, pp. 184–202, Jan. 2016, doi: 10.1111/bjet.12224.

[8] J. Sweller, 'Cognitive load during problem solving: Effects on learning', *Cogn. Sci.*, vol. 12, no. 2, pp. 257–285, 1988.

[9] J. Sweller and P. Chandler, 'Why Some Material Is Difficult to Learn.', *Cogn. Instr.*, vol. 12, no. 3, pp. 185–233, 1994.

[10] J. Sweller, 'Cognitive technology: Some procedures for facilitating learning and problem solving in mathematics and science.', *J. Educ. Psychol.*, vol. 81, no. 4, p. 457, 1989.

[11] D. Shaffer, W. Doube, and J. Tuovinen, 'Applying Cognitive load theory to computer science education.', in *PPIG*, 2003, p. 12.

[12] S. Garner, *Reducing the cognitive load on novice programmers*. Association for the Advancement of Computing in Education (AACE), 2002.

[13] S. E. Volet, 'Modelling and coaching of relevant metacognitive strategies for enhancing university students' learning', *Learn. Instr.*, vol. 1, no. 4, pp. 319–

336, 1991.

[14] T. M. Shaft, ‘Helping programmers understand computer programs: the use of metacognition’, *ACM SIGMIS Database DATABASE Adv. Inf. Syst.*, vol. 26, no. 4, pp. 25–46, 1995.

[15] M. Havenga, B. Breed, and E. Mentz, ‘Metacognitive and problem-solving skills to promote self-directed learning in computer programming: teachers’ experiences’, *SA-eDUC*, vol. 10, no. 2, 2013.

[16] M. Havenga, ‘The role of metacognitive skills in solving object-oriented programming problems: a case study’, *TD J. Transdiscipl. Res. South. Africa*, vol. 11, no. 1, pp. 133–147, 2015.

[17] T. De Jong, ‘Cognitive load theory, educational research, and instructional design: some food for thought’, *Instr. Sci.*, vol. 38, no. 2, pp. 105–134, 2010.

[18] M. Boekaerts, ‘Cognitive load and self-regulation: Attempts to build a bridge’, *Learn. Instr.*, vol. 51, pp. 90–97, 2017, doi: 10.1016/j.learninstruc.2017.07.001.

[19] M. G. Rhodes, ‘Metacognition’, *Teach. Psychol.*, vol. 46, no. 2, pp. 168–175, 2019.

[20] M. Boekaerts, ‘Self-regulated learning: Where we are today’, *Int. J. Educ. Res.*, vol. 31, no. 6, pp. 445–457, 1999.

[21] J. Prather, B. A. Becker, M. Craig, P. Denny, D. Loksa, and L. Margulieux, ‘What Do We Think We Think We Are Doing? Metacognition and Self-Regulation in Programming’, in *Proceedings of the 2020 ACM Conference on International Computing Education Research*, 2020, pp. 2–13.

[22] I. Glogger-Frey, K. Gaus, and A. Renkl, ‘Learning from direct instruction: Best prepared by several self-regulated or guided invention activities?’, *Learn. Instr.*, vol. 51, pp. 26–35, 2017.

[23] R. O. Prates, C. S. de Souza, and S. D. J. Barbosa, ‘Methods and tools: a method for evaluating the communicability of user interfaces’, *Interactions*, vol. 7, no. 1, pp. 31–38, 2000, doi: 10.1145/328595.328608.

[24] R. Prates, S. D. Barbosa, and C. S. De Souza, ‘Using Communicability Evaluation to Compare Interaction Design of Two HTML Tag Editors’, in *Interaction Design: Beyond human-computer interaction*, H. Sharp, Y. Rogers, and J. Preece, Eds. John Wiley & Sons, 2007, pp. 1–21.

[25] M. W. M. Jaspers, T. Steen, C. Van Den Bos, and M. Geenen, ‘The think aloud method: A guide to user interface design’, *Int. J. Med. Inform.*, vol. 73, no. 11–12, pp. 781–795, 2004, doi: 10.1016/j.ijmedinf.2004.08.003.

[26] A. Ploder and A. Eder, *Semantic Differential*, Second Edi., vol. 21. Elsevier, 2015.

[27] S. Grinschgl, H. S. Meyerhoff, and F. Papenmeier, ‘Interface and interaction design: How mobile touch devices foster cognitive offloading’, *Comput. Human Behav.*, vol. 108, Jul. 2020, doi: 10.1016/j.chb.2020.106317.

[28] A. Roberts and J. Bakker, ‘Mobile device login and break-off in individual surveys of Statistics Netherlands’, *Centraal Bureau voor de Statistiek: Discussion Paper*, 2018. https://www.cbs.nl/nl-nl/achtergrond/2018/06/smartphone-en-tablet-respons-bij-personenonderzoeken (accessed May 29, 2021).

[29] J. Nielsen, ‘10 Usability Heuristics for User Interface Design’, *Nielsen Norman Group*, 1994. https://www.nngroup.com/articles/ten-usability-heuristics/ (accessed May 27, 2021).

[30] M. L. Gordon, T. Althoff, and J. Leskovec, ‘Goal-setting and achievement in activity tracking apps: A case study of myfitnesspal’, *Web Conf. 2019 - Proc. World Wide Web Conf. WWW 2019*, pp. 571–582, 2019, doi: 10.1145/3308558.3313432.

[31] M. N. Ismail, N. A. Ngah, and I. N. Umar, ‘Instructional strategy in the teaching of computer programming: A need assessment analyses’, *Turkish Online J. Educ. Technol.*, vol. 9, no. 2, pp. 125–131, 2010, [Online]. Available: http://tojet.net/articles/v9i2/9214.pdf.

[32] A. Klassen, M. Eibrink-Lunzenauer, and T. Gloggler, ‘Requirements for mobile learning applications in higher education’, in *Proceedings - 2013 IEEE International Symposium on Multimedia, ISM 2013*, 2013, pp. 492–497, doi: 10.1109/ISM.2013.94.

[33] J. Campbell, D. Horton, and M. Craig, ‘Factors for success in online CS1’, in *Proceedings of the 2016 acm conference on innovation and technology in computer science education*, 2016, pp. 320–325.

[34] A. Mitrovic and B. Martin, ‘Evaluating the Effect of Open Student Models on Self-Assessment’, IOS Press, 2007. [Online]. Available: https://ir.canterbury.ac.nz/bitstream/handle/10092/2442/12605676_IJAIED531-final.pdf;jsessionid=BD4699D94BC87CD679D0726B6FE43445?sequence=1.

[35] K. Scheiter, R. Ackerman, and V. Hoogerheide, ‘Looking at Mental Effort Appraisals through a Metacognitive Lens: Are they Biased?’, *Educational Psychology Review*, vol. 32, no. 4. Springer, pp. 1003–1027, Dec. 01, 2020, doi: 10.1007/s10648-020-09555-9.

[36] M. Bannert, M. Hildebrand, and C. Mengelkamp, ‘Effects of a metacognitive support device in learning environments’, *Comput. Human Behav.*, vol. 25, no. 4, pp. 829–835, Jul. 2009, doi: 10.1016/j.chb.2008.07.002.

[37] B. Renner *et al.*, ‘Computer-supported reflective learning: how apps can foster reflection at work’, *Behav. Inf. Technol.*, vol. 39, no. 2, Feb. 2020, doi: 10.1080/0144929X.2019.1595726.

[38] B. Renner, J. Kimmerle, D. Cavael, V. Ziegler, L. Reinmann, and U. Cress, ‘Web-Based Apps for Reflection: A Longitudinal Study With Hospital Staff’, *J. Med. Internet Res.*, vol. 16, no. 3, Mar. 2014, doi: 10.2196/jmir.3040.

[39] K. Cornelius, ‘15 Digital tools for student reflection on learning’, Aug. 12, 2016. https://medium.com/@teachrsolutions/15-digital-tools-for-student-reflection-on-learning-65ef12141fb8 (accessed Apr. 28, 2021).

[40] B. S. Thakare, D. Shirodkar, N. Parween, and S. Parween, ‘State of Art Approaches to Build Cross Platform Mobile Application’, 2014. [Online]. Available: www.ijcaonline.org.

[41] B. Frankston, ‘Progressive Web Apps [Bits Versus Electrons]’, *IEEE Consum. Electron. Mag.*, vol. 7, no. 2, Mar. 2018, doi: 10.1109/MCE.2017.2776463.

[42] Ugulino, W., and Ferreira Pires, L., 2021. The Use of Metacognition to Develop Self-Regulated Learning Skills in Students of a Computer Programming Course. In *Engaging, Engineering, Education: Proceedings SEFI 49th Annual Conference* (no prelo, accepted for publication). SEFI: European Association for Engineering Education.

APPENDIX
A. SCREENS OF PLR

This appendix will include several screens of the PLR to give direction to our current vision. As these are early-stage designs, they are subject to change.

A.1 MENTEE VIEWS

The Mentee is a student who has been assigned a Mentor. A Mentor is someone with a great deal of familiarity with the course material who can guide the Mentee in their studies.

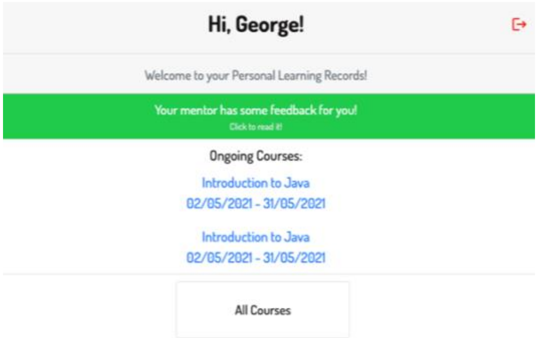


Figure ***A-1. Mentee landing page.***

Figure ***A-2. Mentee course overview.***

Figure ***A-3. Mentee course detail.***

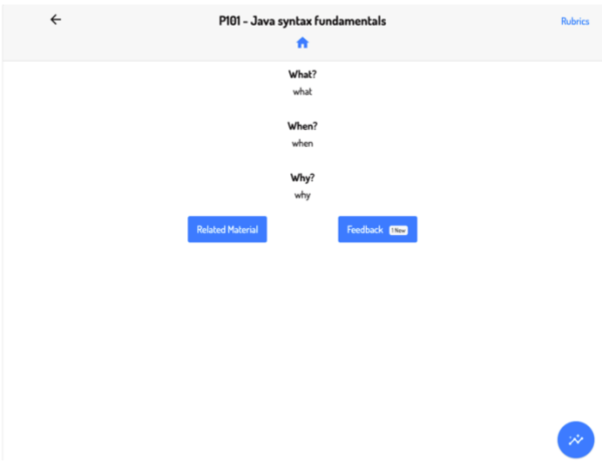


Figure ***A-4. Mentee topic detail.***

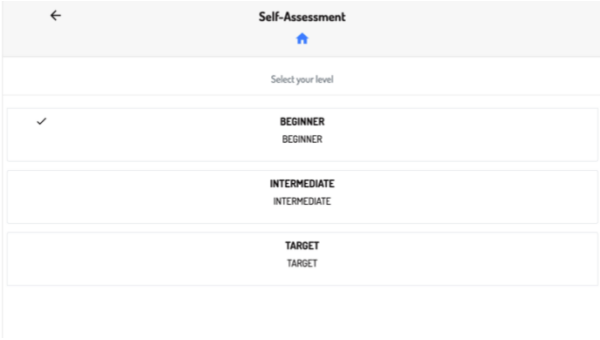
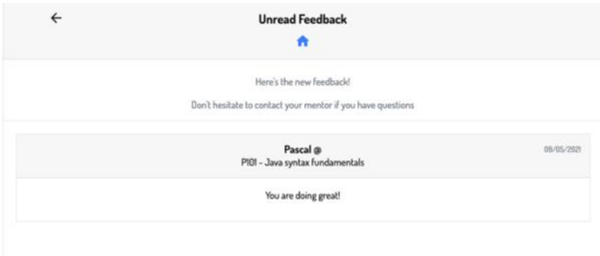


Figure ***A-5. Mentee self-assessment***



. Figure ***A-6. Mentee self-assessment***

A.2 MENTOR/TEACHER VIEWS

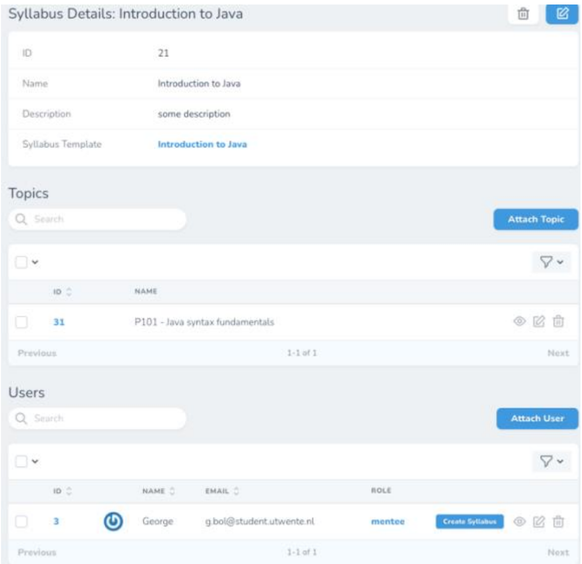


Figure ***A-4. Student statistics for mentors/teachers.***

A.3 DATA MODEL

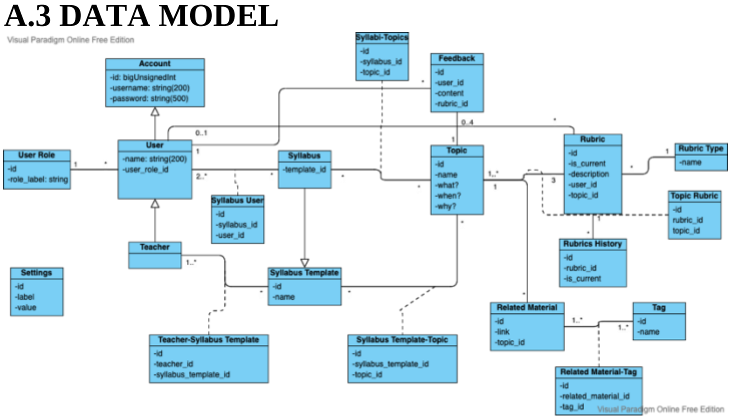


Figure ***A-5. Data model for Personal Learning Records.***