Investigating a learning analytics interface for automatically marked programming assessments

By

Kinsley Ndenge

Supervised by Professor Hussein Suleman

A minor dissertation submitted in partial fulfilment of the requirements for the degree of Masters in Information Technology

Department of Computer Science

University of Cape Town

December 2020

Declaration

This dissertation has been submitted to Turnitin, and I confirm that my supervisor has been shown the Turnitin report. All concerns have also been resolved.

The research was carried out with students at the University of Cape Town, and I did receive ethics approval from the Department of Computer Science before conducting the research.

I declare that the dissertation, hereby submitted by me for the Masters in Philosophy degree at the University of Cape Town, is my independent work and has not previously been submitted by me at another university or faculty. All the sources I have used or quoted have been indicated and acknowledged using complete references.

....

Abstract

Student numbers at the University of Cape Town continue to grow, with an increasing number of students enrolling to study programming courses. With this increase in numbers, it becomes difficult for lecturers to provide individualised feedback on programming assessments submitted by students. To solve this, the university utilises an automatic marking tool for marking assignments and providing feedback. Students can submit assignments and receive instant feedback on marks allocated or errors in their submissions. This tool saves time as lecturers spend less time on marking and provides instant feedback on submitted code, hence providing the student with an opportunity to correct errors in their submitted code. However, most students have identified areas where improvements can be made on the interface between the automatic marker and the submitted programs. This study investigates the potential of creating a learning analytics inspired dashboard interface to improve the feedback provided to students on their submitted programs. A focus group consisting of computer science class representatives was organised, and feedback from this focus group was used to create dashboard mock-ups. These mock-ups were then used to develop high-fidelity learning analytics inspired dashboard prototypes that were tested by first-year computer science students to determine if the interfaces were useful and usable. The prototypes were designed using the Python programming language and Plotly Python library. User-centred design methods were employed by eliciting constant feedback from students during the prototyping and design of the learning analytics inspired interface. A usability study was employed where students were required to use the dashboard and then provide feedback on its use by completing a questionnaire. The questionnaire was designed using Nielsen's Usability Heuristics and AttrakDiff. These methods also assisted in the evaluation of the dashboard design. The research showed that students considered a learning analytics dashboard as an essential tool that could help them as they learn to program. Students found the dashboard useful and had an overall understanding of the specific features they would like to see implemented on a learning analytics inspired dashboard used by the automatic marking tool. Some of the specific features mentioned by students include overall performance, duly performed needed to qualify for exams, highest score, assignment due dates, class average score, and most common errors. This research

hopes to provide insight on how automatically marked programming assessments could be displayed to students in a way that supports learning.

Acknowledgement

I wish to express my heartfelt gratitude to my supervisor, Professor Hussein Suleman, for his guidance and commitment he showed during this process. His feedback and persistence were of immense help.

I would like to thank my friends, Ntsako Mgiba and Chinyama Francis, for their valuable support. Your teamwork was the reason I could complete this dissertation.

To my partner, Rebecca Wanjiku, for her constant support.

To my family, for their never-ending love.

Table of Contents

List of Abbreviations

List of Figures

List of Tables

1. INTRODUCTION

1.1. Background of the Study

For the last ten years, thousands of students at the University of Cape Town (UCT) have been submitting their programming assignments through the automatic marking tool linked to Vula (Sakai[1]), the University's learning management system (LMS). Students receive instant feedback through a report generated by this tool. Submitted programs are repeatedly compiled, executed, and tested (Suleman, 2008). This system has been successful administratively and has improved over the years to better serve the needs of students and University administrators. This research aims to determine if a student-facing learning analytics (LA) inspired dashboard is useful to students when using the Vula automatic marking tool. This research will provide understanding of how feedback from the automatic marker can be better designed to assist first-year University students who are learning to program.

Established in 2011, LA is a relatively new field involved with the measurement, collection, analysis, and reporting of data about learners and their context to optimise learning and the environments in which they occur (Jivet et al., 2018). The data are mainly used for providing feedback to educators during courses and students about their progress. Simple examples of LA are integrated into LMSs as dashboards that indicate progress with various outcomes and highlight problem areas for students.

Automatic marking refers to assessments conducted by a computer. There are numerous ways in which such assessment can be affected, including essay marking by text analysis; online tests, often through multiple-choice and short answer questions; the processing of paper-based forms; and automatic assessments of electronic artefacts (Faniran & Ajayi, 2016). The last category is of most interest to computer scientists as they can have a computer assess computer programs automatically.

The problem domain of this study is the application of LA to the automatic marking of computer programs. The data that can be collected also indicate specific problem areas and trends in learning that can assist students more so than other domains. It

[1] https://www.sakailms.org/

is known that many students use computer-based assessment (CBA) tools in unusual ways, for example, like a compiler or learning to program through trial and error. The signature of student usage can often signal poor habits or difficulty in learning specific concepts to the informed observer.

LA dashboards have been used quite successfully in LMSs within various contexts (Kitto et al., 2016). These can also be seen in massive open online courses (MOOCs) (Díaz et al., 2014). Existing research shows the critical nature of visual displays for sensemaking as we can process large amounts of data better when it is presented in a meaningful way (Schwendimann et al., 2016).

1.2. Motivation for the Study

The number of students admitted to computer science and engineering departments at UCT has increased steadily over the years. This rise has resulted in increased use of the automatic marking tool for marking programming assessments. Despite this tool being very successful administratively and having been improved continuously over the years, some students still have difficulties using it (Suleman, 2008). Fig 1.1 below shows the submission attempts by a learner for a programming assessment.

Assignment	Due Date	Attempts	Submissions
Python Assignment One	2018-03-12T23:55:00	21 / 50	Assignment1.zip [0] Mon Mar 5 13:16:07 2018 [report] [download] [explode] [illegible].zip [0] Mon Mar 12 05:06:01 2018 [report] [download] [explode] [illegible].zip [0] Mon Mar 12 05:30:40 2018 [report] [download] [explode] [illegible].zip [0] Mon Mar 12 05:39:38 2018 [report] [download] [explode] [illegible].zip [0] Mon Mar 12 05:50:48 2018 [report] [download] [explode] [illegible].zip [0] Mon Mar 12 05:52:33 2018 [report] [download] [explode] [illegible].zip [0] Mon Mar 12 11:55:29 2018 [report] [download] [explode] [illegible].zip [0] Mon Mar 12 12:09:06 2018 [report] [download] [explode] [illegible].zip [0] Mon Mar 12 12:24:10 2018 [report] [download] [explode] [illegible].zip [0] Mon Mar 12 12:33:15 2018 [report] [download] [explode] [illegible].zip [0] Mon Mar 12 14:24:05 2018 [report] [download] [explode] [illegible].zip [0] Mon Mar 12 14:54:31 2018 [report] [download] [explode] [illegible].zip [0] Mon Mar 12 15:35:51 2018 [report] [download] [explode] [illegible].zip [35] Mon Mar 12 15:38:15 2018 [report] [download] [explode] [illegible].zip [35] Mon Mar 12 16:09:42 2018 [report] [download] [explode] [illegible].zip [35] Mon Mar 12 17:06:43 2018 [report] [download] [explode] [illegible].zip [35] Mon Mar 12 17:09:25 2018 [report] [download] [explode] [illegible].zip [70] Mon Mar 12 17:12:35 2018 [report] [download] [explode] [illegible].zip [70] Mon Mar 12 17:51:25 2018 [report] [download] [explode] [illegible].zip [70] Mon Mar 12 20:58:54 2018 [report] [download] [explode] [illegible].zip [70] Mon Mar 12 21:03:00 2018 [report] [download] [explode]

Fig 1.1 Sample Output from Automatic Marker

From Fig 1.1, shows that students submit assignments multiple times to the automatic marker. This has an advantage because the learner can iteratively improve on their code and resubmit the assignment (Suleman, 2008). In this case it is clear that, over time, students make corrections in their programs and earn better marks; however, it might be helpful to understand why students are making multiple submissions and how it helps them during this process. The student can see an output from the automatic marker that shows them the expected output of their programs (see Fig 1.2). This enables students to determine where errors originate from and iteratively improve on their submitted code.

```
Trial 2:

Executing program

*****Anagramsets Search*****
Enter word length:
Searching...
Enter file name:
Writing results...
Comparing output

Output not correct

The expected output was:
['opt', 'pot', 'top']
['rot', 'tor']

Your program produced:
['opt', 'pot']
['opt', 'top']
['pot', 'top']
['rot', 'tor']

Contents of EnglishWords.txt:
This word list is derived from the Princeton WordNet. Please refer to
the copyright notice and statements.

00-database-url
     http://www.cogsci.princeton.edu/~wn/
00-database-short
     WordNet (r) 1.7
00-database-long
     WordNet (r): A Lexical Database for English from the
      Cognitive Science Laboratory at Princeton University
00-database-info
     This file was converted from the original database on:
                Sat Jun 23 14:21:23 2001
```

Fig 1.2 Listing Output from Automatic Marker

Although the output from the automatic marker is meant to provide students with some information about their submitted programs, it is often difficult for them to understand and interpret. One reason for this could be because they are just beginning to learn how to program. Students often complain that the automatic marker scores them zero, despite them having run their programs successfully on other integrated development environments (IDEs). This could be very frustrating and discouraging for the learners. Fig 1.3 below shows WhatsApp[2] comments of students in a chat group complaining about the outputs from the automatic marker.

Student 1: q1, q2, q3 appears to be correct, but when i submit, automarker gives 0 out of 100, anyone else getting the same?

Student 2: The output needs to match exactly, including line breaks and characters.

Student 3: I am still getting zeroes for all trials in question 4 despite my code working - I need user input for the command but the auto marker does not allow for that?

Student 4: I must say that Automarker is giving 0 for q4, even though my q4 works 100%.

Fig 1.3 WhatsApp Chat of Students using the Automatic Marker

This study will, therefore, investigate if the implementation of LA inspired dashboards into an existing system for giving students feedback on programming assessments, will result in an improved learning experience.

[2] https://www.whatsapp.com/

1.3. Limitations of the Study

The study is limited to students using the automatic marking tool linked to Vula[3]. It utilises an approach identified by Díaz et al. (2014) for using LA to make sense of data that are generated in online learning environments, which involves building systems that can automatically process the generated data. The study does not make use of a custom system built from scratch. Instead, it utilises a Python library to build and test the usefulness and usability of a prototype LA dashboard. The dashboard designed is not directly integrated into the operational automatic marker but only tested within the Vula LMS to understand how useful dashboards could be to learners.

1.4. Research Question

1) Can an LA dashboard provide students with useful feedback in automatically marked programming assessments?

1.5. Methodology

The methodology is divided into three sections, as discussed below.

1.5.1. Focus Group Interviews

A focus group was conducted with six computer science class representatives. These students were required to sign the informed consent form, which permitted the researcher to record the interview sessions and use the information they provide about their experiences with the automatic marker in the design of a high-fidelity LA dashboard prototype. Senior class representatives were chosen for the focus group due to their experience with using the automatic marker and having a better understanding of some difficulties faced by students while using this tool. Also, a focus group was chosen as a research method because it could stimulate ideation from students based on what other students are saying, thereby leading to more detailed participation and feedback. Before this focus group, a low-fidelity LA dashboard prototype was designed based on existing literature and presented to the focus group participants. The following information was displayed in the prototype: the number of

[3] https://vula.uct.ac.za/portal

submissions allowed, the number of submissions attempted, most common errors, days to the submission deadline, progress trackers, comparisons to other students, scheduling information such as timetabling and hand-in dates, module information, relevant resource links, social interaction, test results and time spent. During the focus group, a prototype was presented to the students so that they can view a draft LA dashboard design. The below questions were asked, and feedback elicited.

1) Is the information currently displayed on the prototype useful to you? Explain.
2) Of the above features displayed on the prototype, which would be useful to you?

1.5.2. Designing the Dashboard

Based on the feedback from the initial focus group interviews, a low-fidelity dashboard mock-up was designed using prototyping tools such as Balsamiq[4]. The second round of focus group interviews was done where the mock-ups were tested again with users and further feedback elicited.

1.5.3 Final Version and Evaluation

Once the mock-up design was refined, a high-fidelity learning analytics dashboard prototype was designed using the Python Plotly package and Dash[5]. Data used for the design were artificially generated using Python libraries such as Numpy and Pandas. Once the dashboard development was completed, it was hosted on a Heroku[6] server where a sample of UCT computer science and engineering students were invited to use the application and then complete a usability questionnaire.

The users were asked to view the dashboard, use the various features of the dashboard, and provide feedback on their perceptions of usefulness and usability of the dashboard by completing the questionnaire.

---

[4] https://balsamiq.com/
[5] https://plot.ly/products/dash/
[6] https://www.heroku.com/

1.6. Thesis Structure

The thesis is made up of five chapters. Chapter 2 presents a broad overview of the existing literature in this research area. It is further broken down to include literature on LA, human-centred design (HCD) and automatically marked programming assessments. Chapter 3 presents the methodology of the research, where HCD methods were analysed to answer the research question. Chapter 4 presents a discussion of the results, which is followed by the concluding chapter (Chapter 5) that summarises the results and findings and proposes possible areas for future research.

2. LITERATURE REVIEW

2.1. Introduction

This chapter begins with a definition of LA dashboards, how dashboards are used by professional software developers, and how this approach can be integrated into the design and use of LA dashboards for students. Design approaches for student-facing dashboards and dashboards designs for LA are reviewed. Studies on how dashboards have been integrated into LMSs are also reviewed. We discuss literature on automatic marking and how it has been used in programming and non-programming assessments, as well as the use of HCD in the development of LA dashboards. Finally, the chapter discusses gaps that were identified when reviewing the literature and concludes with a summary.

2.2. Learning Analytics Dashboards

Learning dashboards are “single displays that aggregate different indicators about learner(s), learning process(es) and/or learning context(s) into one or multiple visualizations” (Schwendimann et al., 2016:2). There have been several definitions for what LA is, but in this study, the definition proposed by Jivet et al. (2018) is taken into consideration. They define LA as collecting, analysing, and reporting of data about learners in the context within which learning occurs. Vast amounts of data, including personal data, interaction data, and academic data has been generated in online learning environments due to widespread use. LA can leverage this data by bridging the gap between data science and learning science to improve educational outcomes (Jivet et al., 2018). LA has the potential to profoundly impact educational practice by creating tools that provide the learner with information about their learning status in an interactive manner (Aleven et al., 2018). Data generated online by students can be fed back to them in a visually appealing way using LA dashboards. From a perspective of human-centred computing, the interpretation of these LA dashboards by those who use them can expose critical design challenges, which are often trivialise. Although a vast array of knowledge does exist about which visualisation interfaces are effective, the practice of implementing known design guidelines is non-trivial (Froese & Tory, 2016). The reason is that usability does not directly correlate to effectiveness from an educational perspective (Echeverria et al., 2018). Although student-facing dashboards have emerged as an easy way of presenting student data, there has been very minimal involvement of students in the process of developing the dashboards (Roberts et al., 2017). The information displayed on dashboards could be used by teachers and learners to inform the decisions they make about teaching and learning. (Corrin & de Barba, 2015). Learners can also use these dashboards as metacognitive tools for reflecting and examining their learning behaviours and learning outcomes (Durall & Analysis, 2014). Dashboards present a user interface (UI) through which data collected can be analysed and displayed from the interaction students have online. This can provide feedback directly to students with information that is related to their engagement and performance, hence helping the students to identify gaps in their performance and make changes to their study strategies (Corrin & de Barba, 2015).

The field of LA has traditionally placed little emphasis on the learner and instead focused on institutions and academics (Kitto et al., 2016). The focus has been on developing tools that could help deliver data about student activities to academics and institutions that will assist them in developing strategies for student retention and also understanding learning behaviours (Corrin & de Barba, 2015). This has led to a disproportionate focus on prediction rather than learning. Dashboards can help bridge this gap by giving students access to their data, which can help them access their performance on learning activities delivered via LMSs. Early warning systems can be incorporated into the design of dashboards. These can assist in providing information to teaching staff about difficulties students might have while engaging with course material. Studies have shown that students use feedback and alerts delivered via dashboards to reflect on their performance and create new and amended study plans (Roberts et al., 2016). Another advantage is the ability for students to instantly see key metrics related to their performance displayed on the dashboard. Apart from this, the feedback presented on an LA dashboard is instant. The efficacy of feedback is vital to student learning as immediate feedback can instantly shape learning behaviours, allowing the student to reflect on what they are doing incorrectly. Research shows that a majority of students are able to interpret the feedback received from an LA dashboard to address gaps in their performance and change the way they study (De Freitas et al., 2019).

2.3. Designing for Learning Analytics

LA is involved with collecting, measuring, analysing and reporting of data about learners and the context within which learning occurs with the aim of understanding and improving learning (Scheffel et al., 2014). The process of LA usually goes through a series of phases, which include the collection of data, doing some pre-processing on the data, analysing the data and doing post-processing of data (Chatti et al., 2012). This process is represented in Fig 2.1 below.

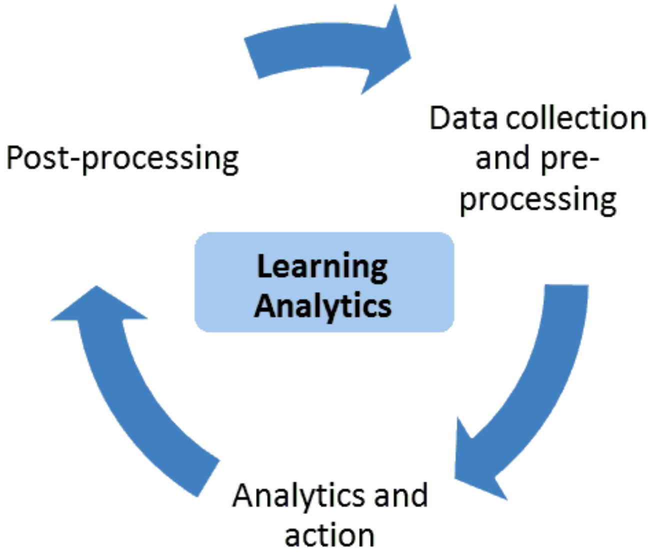


Fig 2.1 Learning Analytics Model (Chatti et al., 2012)

The collection of educational data is foundational to the LA process as, without the right data, the correct analysis cannot be made (Chatti et al., 2012). This step involves the collection of data from various systems and educational environments. Once the data have been collected, it is pre-processed. This can also be referred to as data cleaning, which involves the transformation of data into a state or format that can be used for analytics. These pre-processed data can be used in LA to explore and uncover patterns that can help make the learning experience of students more effective and provide institutions with relevant information to improve the experience of learners. The data can be used proactively to mitigate learning risk, improve engagement and learner performance, thus providing institutions with the ability to improve eLearning courses by gaining valuable insights from learner behaviour (Mothukuri et al., 2017). LA has been used by educational institutions to support student success by using methods such as predictive analytics and machine learning to help identify students at risk of encountering challenges with retention and graduation, leaving/dropping out of a course or dropping out of university (Picciano, 2012). It has also been used extensively to provide feedback and data visualisation. This has led to the improvement of pedagogy. However, studies have found that poor feedback and data visualisation of LA systems can inhibit their use (Ali et al., 2012).

Important questions need to be answered when designing an LA intervention. These include (Wise, 2014):

- How often and when is the right time to use LA in the process of learning and teaching?
- What kinds of analytics should teachers and learners have access to?
- How will the analytics information be used, and what questions should be answered when an analytics system is being accessed?
- How does using analytics fit into the process of learning and teaching i.e., what is the context and how is it used to interpret the information provided by the analytics system?

In general, learners and teachers using LA are more likely to be interested in dashboards when the system helps them make better decisions about the learning event that they are engaged in (Clow, 2012).

2.4. Design Approaches of Developer Dashboards

Beyond the classroom, dashboards will continue to have an impact on the lives of students. Most professional software developers use dashboards to view project status (Ivanov et al., 2018). Dashboard visualisations help software developers view the overall status of projects they are working on, and make individual and collaborative decisions on the projects, which could increase the overall productivity of the developer or development team (Ivanov et al., 2018). Most programmers and development teams consider the awareness of the overall project status and current task as vital aspects for short-term planning during the software development process (Treude & Storey, 2010). On a larger scale, the success of software development depends on how well teams communicate. Due to the complexity of modern software systems, development environments are more advanced and provide better support for programmer awareness.

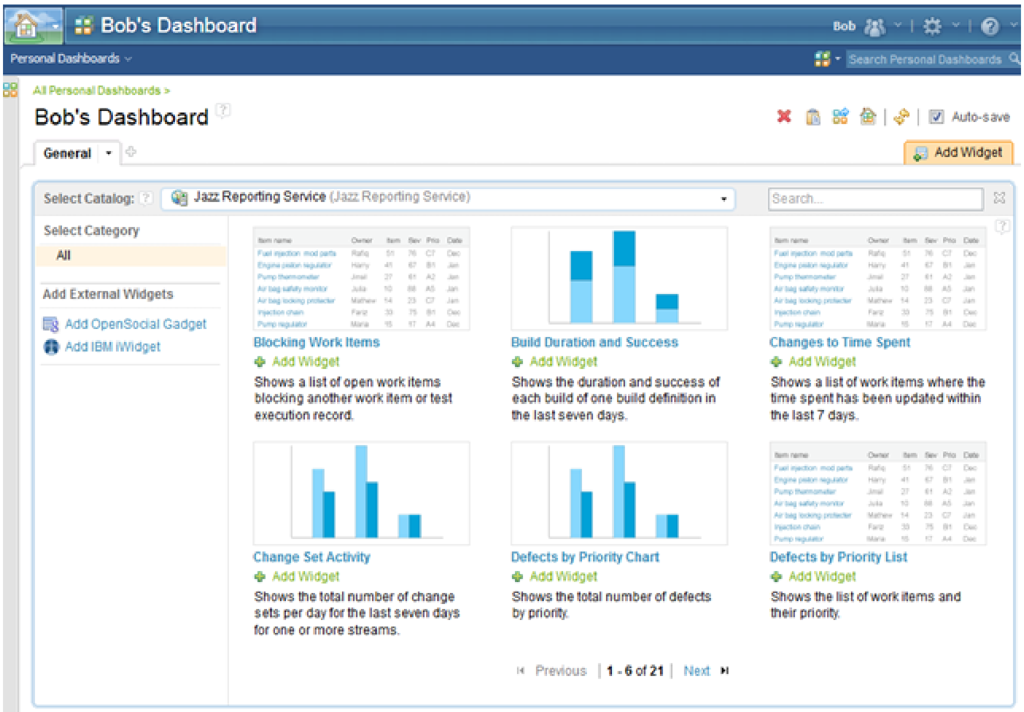


Fig 2.2 IBM's Jazz Software Development Environment

IBM's Jazz software development environment (Fig 2.2) includes the aggregation of data and feeds to help provide awareness to the programmer of low and high-level aspects during the software development process (Finlay et al., 2011). In complex software development, awareness includes being informed of both social and technical aspects related to development, work that is currently being done as well as future work to be done (Treude & Storey, 2010). Awareness is the ability to understand the activities of other people, which provides context for your own activities (Dourish & Bellotti, 1992). Two approaches have been employed in providing awareness on dashboards used by programmers. The first approach is explicit and uses directed messaging. In contrast, the second approach uses a "strong" notion of roles and activities to help convey information to the group about an individual's actions and plans, i.e., providing shared results and feedback from the activities of individuals to the group through a shared workspace (Dourish & Bellotti, 1992). The above principles, although related to dashboard systems used by larger development teams, can be incorporated into our LA dashboard to provide students with greater visibility of the programs submitted and other key metrics that can improve their learning and provide greater awareness of their development as they learn to write good code.

2.5. Design Approaches of Student-Facing Dashboards

Several approaches have been used in the implementation and research of student-facing dashboards in computing education. In a comprehensive literature review of learning dashboard studies (Schwendimann et al., 2015), researchers identified a need for studies that systematically compare different dashboard designs in authentic learning settings. Researchers have been able to collect useful learning data for the purpose of LA. Example data includes time on task, types of errors, constructs used, debugger used, lines of code, types and length of methods written. When these errors are received hints or alerts are automatically sent to assist both learners and teachers in making learning adjustments based on available data (Olivares, 2015). Others have collected compiling logs from students and used these to locate different types of syntax errors (Fu et al., 2017) or used data sources outside the traditional LMS such as Twitter to design dashboards that assisted teachers to facilitate discussions, as well as introduce relevant outside resources to their teaching (Gruzd & Conroy, 2018). In one study (de Quincey et al., 2016), students were asked to propose preferences for data that should be collected and displayed on an LA dashboard. The features that were chosen included representation of attendance, assessments, representation of activity, progress metres, information that compares performance with colleagues, scheduling information such as timetabling and hand-in dates, module information, relevant resource links, and simpler communication methods. Through such research, recommendations have been provided to students on how to improve their learning. With research in the field of LA expanding, there is a need to move from research that simply measures student use of a dashboard to research that show how students respond to information presented on a learning analytics dashboards (Kitto et al., 2016). The challenge for LA remains the establishment of a plausible relationship between models that are derived from digital data, which is quantifiable, and learning, which is a complex socio-cognitive endeavour (Kitto et al., 2018). This can be achieved by focusing research on dashboards that creates awareness, reflection, sensemaking and which can lead to a change in behaviour. LA dashboards should enable learners to create more awareness about the course they are studying, provide an opportunity for reflection, sensemaking, and ultimately, positively influence the behaviour of the student (Verbert et al., 2014). These visualisations should create new meaning and insight to help students in their learning process. This is graphically depicted in the LA process model (Verbert et al., 2014) provided in Fig 2.3 below.

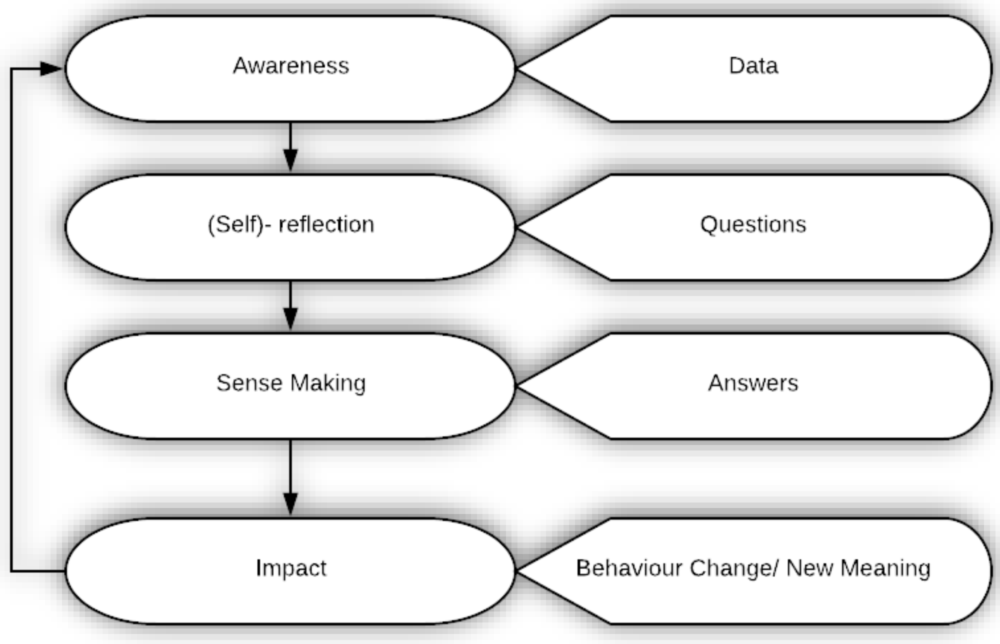


Fig 2.3 Learning Analytics Process Model (Verbert et al., 2014)

In the LA process model, when learners and teachers face an LA dashboard or visualisation, they go through several awareness/sensemaking phases where they try to make sense of the visualisation (Verbert et al., 2014). Learners formulate questions based on the presented data and assess the usefulness of the data in addressing the questions they have. Learners can respond to their formulated questions with new insights, like adapting their behaviours.

Data traces from student logs in programming assignment submissions can be used to provide a dashboard visualisation that creates awareness to the learner and enables sensemaking, thereby facilitating learning. Data collected in the development of learning dashboard applications include artefacts produced when using the application, time spent during its use, social interaction within the applications environment, resources used, and results of test or exercises. The most used constructs for dashboard design are the social interactions and artefacts generated during use (Verbert et al., 2014).

The evaluation of these dashboards has typically involved the determination of their usability, usefulness, gains in efficiency and effectiveness (Verbert et al., 2014). Measuring usability and usefulness could be done by eliciting feedback from learners. Gains in efficiency and effectiveness need a controlled environment and longer time to measure. This research will focus on measuring the usability and usefulness of the learning dashboard implemented for automatically marked programming assignments.

2.6. Human-Centred Design of Learning Analytics

Presenting data generated online in the form of a dashboard requires the extraction of information from disparate sources, which might differ from the interpretive needs of the learner (de Quincey et al., 2016). Making information available to stakeholders is only part of the analytic process. Presenting this information to students in a way they understand could help tackle the inability of the learner to access a comprehensive view of their current learning activity, which is even more critical. Presenting this information to students requires writing computer programs to collect data. In his book, Wirth (1975) defined programs as a combination of data and algorithms. Computer programs act on data streams as input to produce an output. This data can be generated in an educational setting by students using an LMS. A user-centred approach to LA design puts students at the centre of the design process. This does not only reduce undesirable issues that are related to accessibility and usability but also ensures that the data represented in the design is what the student wants (de Quincey et al., 2016). Learning is a complex process, which usually occurs within a particular context. The data generated by learners embody the complexity inherent in the relationships within a learning context. These relationships include the learners' relationship with themselves, the interrelationships between the different dimensions of learning power, both intrapersonal and interpersonal relationships with other learners, and the inter-contextual relationship between learners and their context (Deakin Crick et al., 2015). Within this context, the learner is the driver of change. Understanding the learner requires understanding what they do, i.e., studying the artefacts they produce as they engage in the process of learning.

A human-centred approach puts the learners at the centre of this change. It builds processes, i.e., temporal and lateral connectivity of learning that supports the flow of information rather than inhibits them. By providing a close alignment between educational visualisations with the intended learning design, researchers have been able to propose the concept of storytelling as an approach to presenting and explaining student educational data (Echeverria et al., 2018). Data storytelling supports sensemaking and guides students and teachers by communicating insights through a combination of data, insights, and narratives. In their research, Echeverria et al. (2018) proposed the following data story design elements that can be used to communicate data:

- A prescriptive title with a message, which clearly communicates the intention of the visualisation. The same data can communicate multiple stories. Hence, a good title serves to provide insight on what is communicated on the current visualisation.
- Using lines and other artefacts to highlight specific data series, i.e., a line for a specific visualisation could be made thicker.
- Specific data points could be highlighted by removing irrelevant data points.
- Decluttering the visualisation by removing grids, limiting the number of different colours used, de-emphasising data that is not the centre of a data story.
- Adding narrative text to data visualisation to make it easier for students and teachers to understand the data.

They called the data storytelling approach a learning design-driven model or an LA process model (see Fig 2.4).

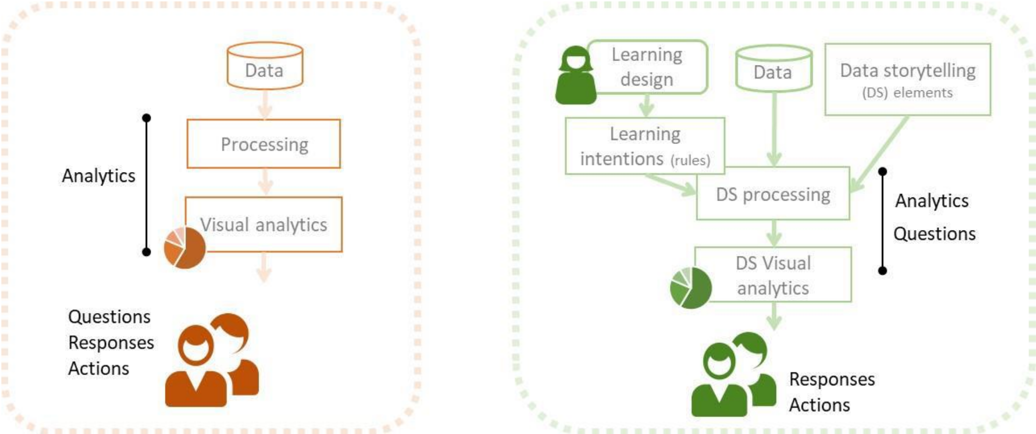


Fig 2.4 Conceptual models. Left: Conventional data-driven visual analytic approach. Right: Learning design-driven, data storytelling approach to support sensemaking (Echeverria et al., 2018).

In Fig 2.4 above, the visualisation on the left shows a traditional data-driven approach to visualisation where raw learner data are used to create visual analytics, i.e., predictive models. In the LA process model on the right, user questions are used to enable the modification of critical elements of visualisation. In this approach, educators' learning intentions are translated into storytelling elements that highlight insights from the data. The teachers' pedagogical intentions are implicitly or explicitly used in the learning design. Rules can be derived from these intentions and the rules are readable by the data processing system for LA (Echeverria et al., 2018). In general, the process of presenting analytics data in a way that impacts learning involves a clear understanding of learner and teacher needs.

In their research, Fukuzumi et al. (2017) used an HCD process to develop software that can be used to understand user context better. They proposed the following six steps, which will be further expanded upon in Chapter 3 of this study.

- Plan the process of HCD.
- Understand the context where this process will be used.
- Specify the user needs and their requirements.
- Produce a design solution that meets user requirements.
- Evaluate the design against the user requirements.
- Evaluate if the solution that was designed meets the user requirement.

These steps are demonstrated in the relationship diagram as graphically depicted in Fig 2.5 overleaf.

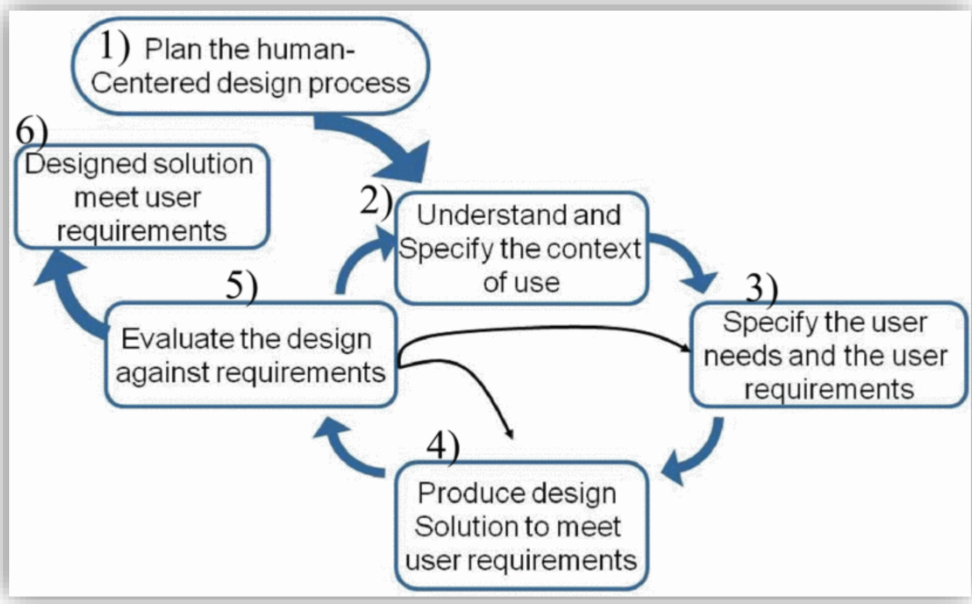


Fig 2.5 Relationship amongst HCD Activities (Fukuzumi et al., 2017)

HCD as a method produces systems and products with high usability for users. However, although there are guidelines for measuring usability, some research has highlighted difficulties faced by software engineers while designing activities that measure usability (Seffah et al., 2004). Consequently, there is a need to streamline how the HCD process is applied to software development (Fukuzumi et al., 2017). A usability evaluation test developed by Fukuzumi et al. (2017) made use of the following criteria: efficiency, errors, ease to learn, and ease to memorize (see Fig 2.6).

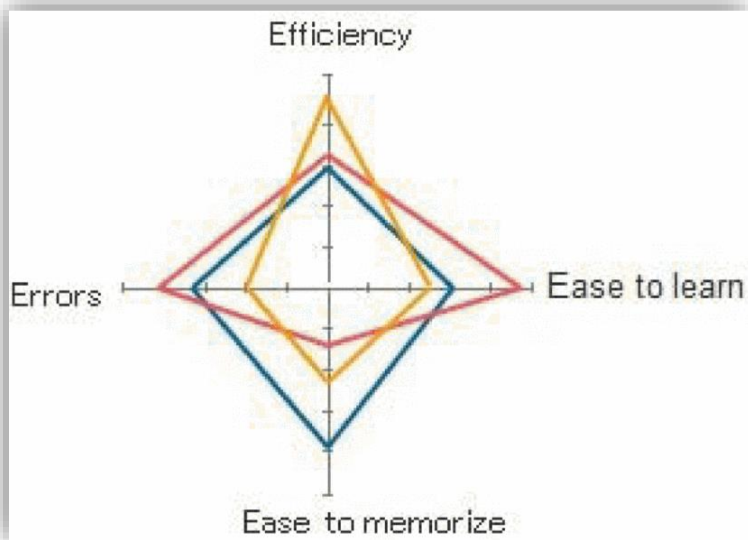


Fig 2.6 Example of a Usability Evaluation Test (Fukuzumi et al., 2017)

Fig 2.6 shows the four points of view from which the results of evaluations can be analysed. For each point in the radar chart, weight and coefficient were set for use in analysis which, in turn, assisted in determining which aspect of the usability needed to be improved.

As demonstrated in Fig 2.3, an LA tool should create impact by enabling sensemaking. It should allow learners to better plan their activities as well as reflect on them, thereby increase learners' awareness of their own actions and the learning process (Scheffel et al., 2014). Awareness of one's situation is a pre-requisite for deciding and performing a task. As Bolton (2010) noted, reflection promotes insights about previously unnoticed details and elicits changes in learning behaviour. Although awareness and reflection are important constructs, they are not easy to evaluate because of a lack of standards to which tools that aim to foster such qualitative values, can be measured against (Scheffel et al., 2014). Scheffel et al. (2014) study used a group concept mapping (GCM) approach to solve this problem. They created a visualisation of ideas from a target group which was done using analytic and data interpretation methods. The output was maps that guided the planning and assessment of issues identified among the target group.

2.7. Computer-Based Assessments

Information and communications technology (ICT) has made possible a move from paper-based assessments (PBA) to CBA (Faniran & Ajayi, 2016). Many professional examinations such as Association of Certified Chartered Accountants, Cisco Networking Engineer, and the Cisco Certified Network Associate examinations, moved from PBAs to CBAs (Faniran & Ajayi, 2016). CBAs have reduced the time and cost associated with administering and grading such examinations. The benefits associated with CBA have proliferated its adoption by universities and other government/multinational organisations. The adoption of CBAs has also created the need for highly effective testing programs to meet this demand (Faniran & Ajayi, 2016). Apart from professional examinations, MOOCs, which are usually delivered by elite universities over the Internet, are driving the adoption of CBA tools. Most MOOCs are not taken for credit purposes; hence the need for summative assessments is unnecessary. However, continuous formative assessment is necessary to enable the learners to evaluate their learning and improve their learning (Pieterse, 2013). Apart from the examples above, traditional universities are experiencing a rapid increase in student numbers and a constantly increasing class size, necessitating the search for effective ways to teach software skills within the prevailing resource constraints (Ngo-Pham et al., 2005). Institutions of higher education, such as UCT, have also implemented the use of CBAs in some courses. One such case is the use of an automatic marking tool used with the Vula LMS to mark and grade programming exercises. Automating the process of marking programming assignments presents benefits to both students and teachers (Suleman, 2008) in the following ways:

- The workload for teachers with large classes is reduced dramatically.
- A consistent set of rules is used to mark all assignments.
- All marks can be fully explained.
- Administratively, an audit trail of all assessments is kept electronically, including submissions and marking.
- Due to the thorough nature of assessments, which mimics unit testing techniques, students also learn to program more professionally.
- Automation in grading workflow decreased time spent on grading and providing feedback. This can be important for novices who will benefit from the early clarification of programming concepts (Pieterse, 2013).
- Automation also makes it possible for teachers to give more formative assessment tasks to students, hence promoting learning through more practice (Pieterse, 2013). Unlimited submissions enabled through the use of the automatic marker can also support iterative learning (Suleman, 2008).

Apart from the listed advantages of implementing CBA, more stringent requirements are employed in automatic marking as opposed to human marking since partial marks cannot be awarded if the program does not pass the test. Researchers at the University of Nottingham (Mansouri et al., 1998) found that continuous assessment enabled by automatic graders prepared students better for the end of semester project. This also gave teachers enough time to spend on assisting students rather than marking. Although there are positive aspects to automatic marking, researchers at the Open University found that automatically marked grades tend to be lower than human marking (Thomas, 2003). Other researchers (MacWilliam & Malan, 2013) found that automating the grading process reduced the time teachers spent on providing feedback and helping students online or in person. It also reduced staff burnout. Lecturers were able to spend more time online providing asynchronous text feedback to students, although the number of students who do not read such feedback increased over time. However, researchers also found that most students prefer online feedback (Dalgarno et al., 2007). Additionally, most students find electronic systems easy to use. A strong correlation has been reported between the introduction of online feedback mechanisms and higher exam scores have been found to be strongly correlated (Heaney & Daly, 2004). The results from these studies further strengthen the need for further research not just on automating the grading process but also streamlining the feedback process in a way that reduces rather than create friction in the students learning experience. Automatic marking systems have been extensively used to mark code, text, and diagrams at universities.

2.8. Automatic Marking Systems in Computing

Automatic marking implies the use of automated, comprehensive testing techniques which mimic unit testing (Suleman, 2008). Assignment submissions go to a computer program, which marks and provides feedback. Automatic marking tools have been used for the assessment of programming code (Blumenstein et al., 2004; Ngo-Pham et al., 2005; Suleman, 2008), entity-relationship (ER) diagrams (Thomas et al., 2005), and 3D modelling diagrams (Sanna et al., 2012).

2.8.1. Automatically Marking Code

Computer-based marking systems have been used to mark computer programs. Such systems have been shown to significantly reduce the burden of marking from lecturers without any adverse effect on students (Ngo-Pham et al., 2005). While some institutions make use of multiple-choice tests to assess students' programming skills, it can be argued that such tests do not thoroughly evaluate students' ability to develop software segments that can run correctly. A system that automatically marks programs gives teachers the ability to constantly test the programming skills of students, especially in large class settings and without the burden of marking. In their design, Ngo-Pham et al. (2005) used a real-time marking program that compiles and executes software uses the criteria set by the examiner to mark. This was used in an examination setting. Their system is designed as a client-server system, which supports many clients simultaneously. The modular software design structure supports flexibility, which allows addition of components to the existing core structure based on examination needs. Other researchers such as Blumenstein et al. (2004) proposed a generic automated marking environment (GAME) that can mark student assessments written in several programming languages. The system has marker modules that are customised for every programming language. New marker modules can easily be added to extend the functionality of the system. Another automatic marking system (Suleman, 2008) used scripts called graders that are used to mark the assignments. An extensible markup language (XML) configuration file stores information about how the assignment will be marked (see Fig 2.7).

```
<markingguide>
   <question>
      <compile>javac Question1.java 2&gt;&amp;1</compile>
      <run>java Question1 &gt; testoutput.out &lt; testinput.in 2&gt;&amp;1</run>
      <check>
         <file1>testoutput.test</file1>
         <file2>testoutput.out</file2>
      </check>
      <marks>20</marks>
      <trial>
         <file name="testinput.in">
5

         </file>
         <file name="testoutput.test">
Enter the height of the triangle:
    *
   **
  ***
 ****
*****

         </file>
      </trial>
      <trial>
         <file name="testinput.in">
10

         </file>
```

Fig 2.7 XML Configuration File (Suleman, 2008)

The XML files have question and trial nodes that provide the information used to mark the submitted assignments. Each question XML node defines commands that compile and run the program. The node also defines files that will be used for comparison to ascertain correctness. Further, trials are defined, and they specify files that will be created in the system. These files are stored by the automatic marker, which then runs the commands, checking specific files for equality. Marks for each segment are specified in the question and trial nodes. Only correct output is allocated final marks. The system also gives a level of control to the course lecturer, who sets the number of times the assignments can be submitted. Due dates for submitting the assignment are set separately in the LMS (Sakai) whereby students might be automatically subjected to a late penalty if submissions are late.

Security and quality are essential in automatic marking systems to ensure the assessments are of high quality. Several ways are proposed to ensure the quality of automated assessments (Pieterse, 2013):

- Security is key as the system must be secure enough to safeguard against malicious programs.
- It should be easy for students to use.
- The system must be versatile enough to specify a test case and expected solutions. It must support the creativity of the instructor when designing test cases.
- Leniency in student programs evaluation. It should allow for variations in the formatting of output.
- An option for instructors to provide qualitative feedback regarding the output of submitted programs.
- An option to show statistical information about programs uploaded to the automatic marker. Students can learn by knowing the common errors that have occurred during program uploads and can be motivated with knowledge about how they performed in relation to their peers.

Most of the recommendations stated above have been considered in the design of the automatic marking tool; however, students still face difficulties using the system.

The automatic marking systems described so far use a client-server architecture. Other researchers have successfully used automatic marking systems on standalone flash disks (Li et al., 2010). Their system uses a flash disk inserted on a single computer to display a user interface where a program can be executed.

2.8.2. Automatically Marking Text and Diagrams

In their paper, Thomas et al. (2005) present an approach for computers to understand ER diagrams. They explain how the approach can be applied to automatically mark students' attempts at drawing such a diagram. This is a particularly difficult approach to automatic marking because traditionally, much of the activity on diagrammatic reasoning has been on the use of precise diagrams, e.g., mathematics proof signs like delta (∂) and visual query interfaces to geographic information systems. Most student diagrams, like ER diagrams, are imprecise and with malformed features. In their approach to automatically marking diagrams, Thomas et al. (2005) use five stages to understand diagrams submitted by students. These include segmentation, assimilation, identification, aggregation, and integration:

- Segmentation and assimilation translate a raster-based image into a set of diagrammatic primitives such as boxes, text, and lines. Each of these objects is separate with unique attributes which include special coordinates.
- The identification phase uses domain knowledge to identify what the researchers call minimal meaningful units (MMUs).
- In the aggregation stage, MMUs are combined into higher-level abstract features, which are used by the integration phase to look for meaning in the diagram by comparing MMUs to a specimen solution.

Automatic marking has also been applied to text marking (Alikaniotis et al., 2016). Researchers at Oxford used natural language processing where a neural network model was trained to score text consistently and automatically. Human experts were used to manually engineer the predictive features of these models to achieve good performance. The trained model was able to learn the extent to which specific words in a sentence contribute to a text's score.

2.9. Summary

Although LA is a new field, it has attracted numerous research with the primary goal of understanding how the vast amounts of data generated in online and blended learning environments can provide better insights about learners. Such insight would be used make the learning experience better. Much of the research, however, has been a focus on LA as a contingency and intervention support tool where course administrators could look at learning patterns and identify students at “risk” either because they are not accessing crucial content or achieving certain minimum scores in the assessments. There have also been promising work done on the implementation of student-facing LA dashboards, where students can view progress, receive real-time feedback, set goals, and reflect on their learning progress. Results from studies have shown that purpose is a key driver in learning and a good LA dashboard design will create an impact by making the students aware of what they are doing and helping them self-reflect on the task at hand. However, with the varied context from which learners come from, it becomes important to understand these learners before designing learning interventions to assist them in sensemaking. A human-centred design methodology can assist us in achieving this goal by providing a platform where we can better understand the learners and design LA dashboards that most closely meets their needs.

In the section that follows (Chapter 3), the methodology used for carrying out this research, the methods used for collecting data, research samples, and tools is discussed.

3. RESEARCH METHODOLOGY

3.1. Introduction

The study set out to investigate if an LA inspired dashboard can provide students with useful and usable feedback for automatically marked programming assignments. This feedback could be helpful to students who are programming beginners and who might have difficulties writing good quality code.

The literature review shows that dashboards should provide an opportunity for awareness, sensemaking, self-reflection, and impact. The purpose of designing an LA inspired dashboard is not just to have a beautifully designed visualisation tool. These dashboards should fit into the broader purpose of achieving a particular learning outcome (Deakin Crick et al., 2015).

This chapter discusses the procedure used to conduct this research, namely the tools, methodology, and associated Python libraries that were used to design the LA inspired dashboard. It also discusses how we evaluated the dashboard design by carrying out a usability evaluation employing Nielsen's Usability Heuristics (Nielsen, 1994) and AttrakDiff (Walsh et al., 2014). The research demonstrates the application of a user-centred design methodology to design a dashboard and subsequently, evaluate the design.

3.2. Research Objectives

The main aim of the research was to investigate if an LA inspired dashboard can provide students with useful feedback in automatically marked programming assessments. Currently, computer science students use the automatic marker, a tool linked to the Sakai (Vula) LMS to submit programming assignments. Once assignments are submitted, students receive feedback on their submitted programs. The automatic marker has a dashboard, which shows the score students obtained for each program submitted. The feedback also shows the expected output of the programs (see Chapter 1, Fig 1.2). Students can use this output to iteratively improve on their submitted code.

Despite these features, students have complained about the interface where submitted program outputs are displayed, often complaining that the automatic marker scored their programs incorrectly.

This research investigates the possibility of using an LA inspired dashboard where the output from submitted programs can be displayed to provide further context to the students on their submitted programs.

3.3. Research Process

This study utilises user-centric methodologies by employing focus group interviews to understand the users and develop a high-fidelity LA inspired prototype dashboard. This prototype was tested using first-year computer science students at UCT. An initial one-hour focus group interview was conducted with second-year computer science class representatives at UCT to understand users' needs. During this session, the feedback was elicited from the users by asking prepared interview questions and probing further based on the responses received. The focus groups allowed the researcher to understand learner needs and perceptions of the quality indicators of LA (Scheffel et al., 2014). Unstructured questions during the focus group also allowed us to ask probing questions on issues where more clarification was needed. Maxwell (2005) pointed out that an unstructured research approach – in contrast to a structured approach – allows the researcher to focus on a phenomenon that may differ from others and require individually tailored methods. After the first round of focus group interviews was completed, a dashboard sketch was done (see Fig 3.7) based on student feedback. This sketch was later presented to the focus group, and more feedback on features was received. Mock-ups were then designed based on the feedback received from the focus group participants. These mock-ups were iteratively refined based on student and research supervisor feedback until a final version was produced (see Fig 3.10 to 3.15).

After the mock-up designs were completed, a high-fidelity prototype of the dashboard was developed using the Python programming language. Other software packages that were used include Time, Dash, Numpy, Plotly, and Pandas. These research tools are discussed in detail in Section 3.4.

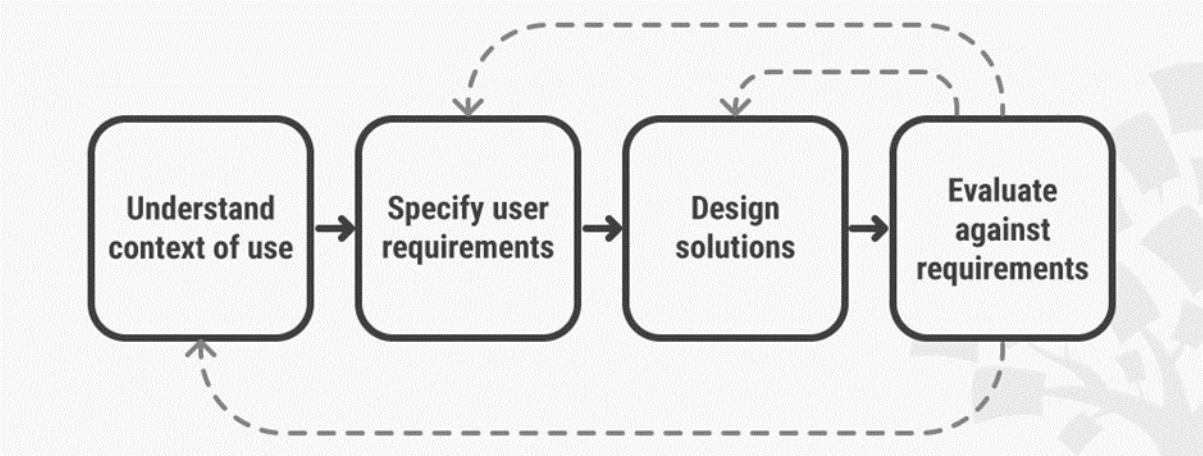


Fig 3.1 User-Centred Design Process (Source: Interaction-Design.org)

Fig 3.1 above is a diagrammatic representation of the user-centred design process. The first step is to understand the users' context. In our study, the needs of first-year students studying computer science at UCT was investigated. Based on these needs, the researcher was able to specify design requirements by conducting focus group interviews. Feedback from these interviews enabled us to design a prototype, which was then evaluated with the students.

3.4. Research Tools

After completing focus group interviews and getting input from research participants, the research focus turned to what tools would be used for the study.

Below follows a description of the tools used during the design phase.

3.4.1. Balsamiq

Balsamiq[7] is a graphical user interface mock-up and website wireframe application that makes prototyping easy by using pre-built widgets and a What-You-See-Is-What-You-Get (WYSIWYG) editor. Balsamiq was used for this study because it allows the easy creation of low-fidelity wireframes that reproduce the experience of sketching on a notepad or a whiteboard. This is important because creating nice-looking prototypes early in the research process could prevent the research participants from providing good input if they feel the researcher has already put in a lot of work to design the prototypes. The focus of the prototype is, therefore, on structure and content and not

[7] https://balsamiq.com/wireframes/

on the details of colour. This allows us to produce mock-ups easily and quickly that can be tested with users and new iterations developed with minimal time constraints. Fig 3.2 shows the LA inspired dashboard interface within the Balsamiq user interface.

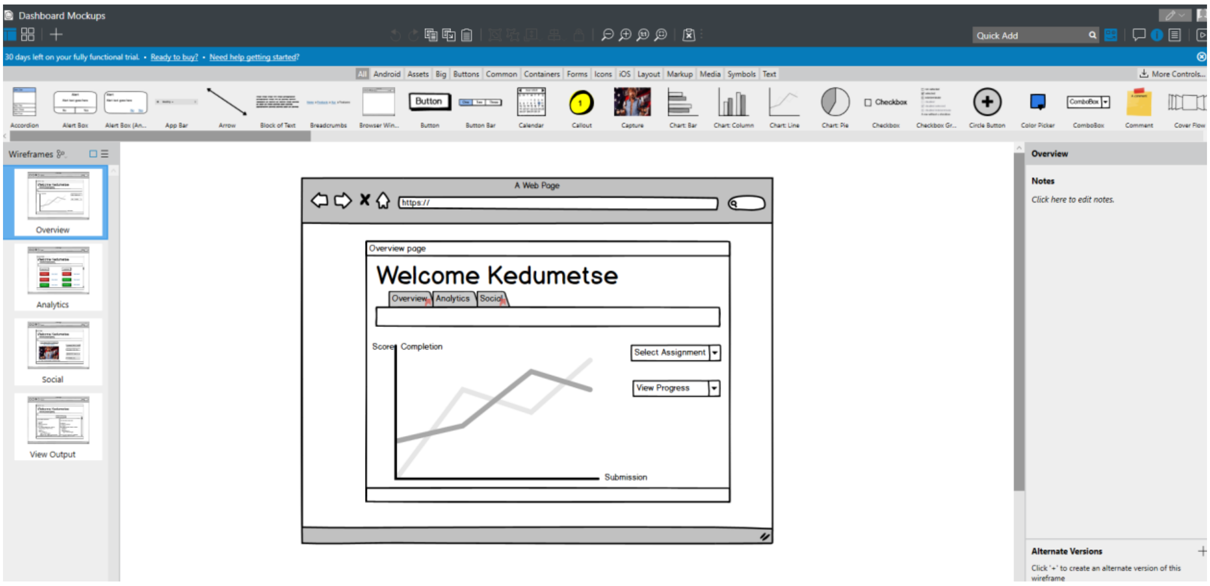


Fig 3.2 Screenshot of the Balsamiq User Interface

3.4.2. Python Libraries and Frameworks

Dash is a framework based on Python, and it is used to build and develop Web applications. It is built on top of other JavaScript (JS) frameworks such as Plotly.js, React, and React.js. These tools make it easy to build highly scalable dashboards using the Python language. Dash is open-source and the Web apps created can run on a Web browser. Several packages that enabled us to build a high-fidelity prototype dashboard using Dash and Python were installed. Fig 3.3 (see overleaf) shows some of the packages imported when building the application.

```
"""
A simple learning analytics dashboard high fidelity prototype
for programming assignment feedback.
"""

#Import Python libraries
import dash
import time
import dash_core_components as dcc
import dash_html_components as html
import plotly.graph_objs as go
import dash_bootstrap_components as dbc
from dash.dependencies import Input, Output, State,
ClientsideFunction
import numpy as np
import pandas as pd
```

Fig 3.3 Code Snippet from Dashboard Python Libraries Import

From the list of imports:

The “time” module is used by the app to provide various time-related functions.

Dash core components are components from the Dash library. These are shipped with Dash and provide an easy way to create interactive user interfaces. With this component, a dropdown can be easily coded using the code in Fig 3.4.

```
import dash_core_components as dcc

dcc.Dropdown(
    options=[
        {'label': 'Assignment 1', 'value': 'Ass1'},
        {'label': 'Assignment 2', 'value': 'Ass2'},

    ],
    value='Ass1'
)
```

Fig 3.4 Dash Core Components Code Snippets

Dash Hypertext Markup Language (HTML) components is a Dash Web application framework that provides pure Python abstraction around HTML, Cascading Style Sheets (CSS) and JS. It makes the development of a dashboard easier because instead of writing HTML or using an HTML templating engine, you can just use Python structures to compose layouts. Fig 3.5 (see overleaf) provides an example of a simple HTML structure.

```
import dash_html_components as html

html.Div([
    html.H1('Hello World'),
    html.Div([
        html.P('Here is a smiple learning analytics
dashboard'),
        html.P('You can view outputs for your submitted
programs here)
    ])
])
```

Fig 3.5 Dash HTML Components Code Snippet

The Plotly graph objects module contains an automatically generated hierarchy of many Python classes. The term “graph objects” refers to instances of these classes, which have methods to easily manipulate them, e.g., update_layout(). These methods are generally used to display visualisations on a page (see Fig 3.6).

```
import plotly.graph_objects as go


fig = go.Figure()
fig.update_layout(legend_title_text = " Assignment Score")
fig.update_xaxes(title_text="Assignment Number")
fig.update_yaxes(title_text="Total Marks")
fig.show()
```

Fig 3.6 Plotly Graph Objects Code Snippet

Pandas[8] is a powerful and easy to use open-source Python library for data analysis and manipulation, while Numpy[9] is a fast and versatile Python package widely used

[8] https://pandas.pydata.org/
[9] https://numpy.org/

for numerical computing. In this research, Numpy was used it to generate random data used for creating visualisations for our high-fidelity prototype LA inspired dashboard.

3.5. Research Sample

The research was done using two different groups of participants. The first group of participants was used for a focus group interview to understand user needs, while the second group of participants was used to evaluate the usability and usefulness of a high-fidelity prototype dashboard. The focus group participants were selected from a group of class representatives, while the second group was selected from first-year computer science students at the School of Information Technology[10], UCT. Both groups of students have programming courses as part of their degree requirements and are required to complete a formative assessment whereby programming assignments are submitted on Vula through the automatic marker. Students get instant feedback from these assignments (see Chapter 1, Fig 1.2), which enables them to compare the output from their submitted assignments to the expected outputs.

The focus group participants were chosen for the following reasons:

- As second-year class representatives, they are familiar with Vula and the automatic marker. This is because they have previously used these tools during their first year of study and have developed a better understanding of how the tool works.
- Class representatives are familiar with some of the difficulties the new students face. This experience means that they are better suited for the focus group as they can contribute ideas based on their experiences. They can provide rich feedback, which will assist in the design of a prototype.

The second group of participants was used to test the usability and usefulness of the prototype. They were chosen for the following reasons:

- As first-year students, they do not have much experience with using the automatic marker, and testing a prototype on them can provide a better understanding of the usability and usefulness of the LA interface design.

[10] http://www.sit.uct.ac.za/

3.6. Focus Group

To prepare for the interviews, an LA interface was drawn by hand and used during the focus group interview. The initial sketch of the dashboard (see Fig 3.7) was based on existing literature on dashboard designs for LA. The initial focus group lasted about an hour. An invitation was sent out to eight students; however, only five showed up for the session. This session was recorded. Students were asked the following two questions during which more probing was done to elicit feedback:

1) Is the information currently displayed on the prototype useful to you (see Figure 3.7)? If so, explain why.
2) Of the features displayed on the prototype, which would be useful to you and why (see Fig 3.7)? Any more additions?

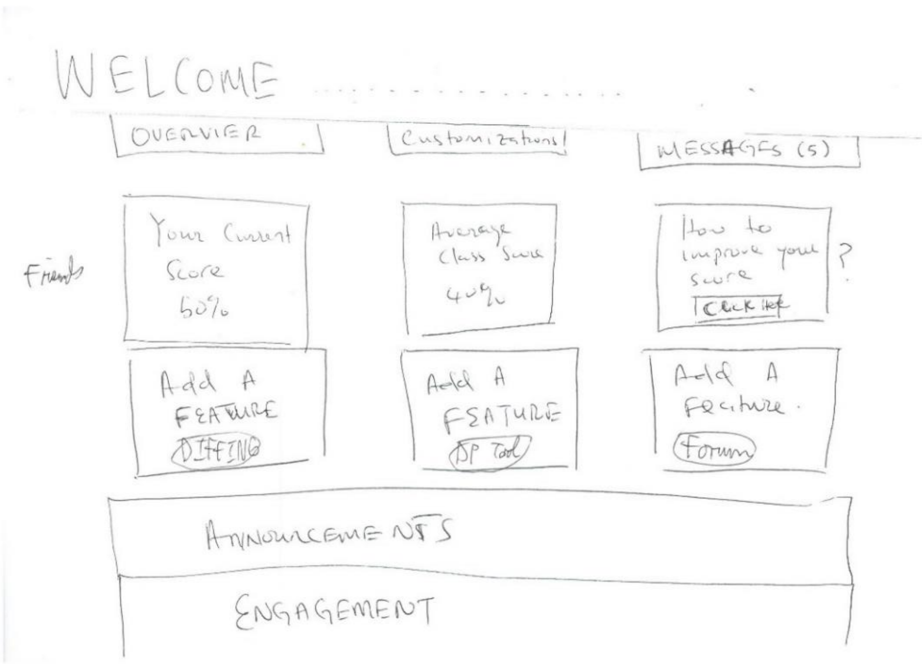


Fig 3.7 Dashboard Sketch used during Focus Groups

3.6.1. Feedback to Questions during Focus Groups

In response to Questions 1 and 2 above, all the computer science class representatives said that the information currently displayed on the prototype was useful. Students were familiar with our initial rendition of the dashboard prototype and some of the features that were sketched on the prototype. During the focus group, students asked if we are going to be changing the automatic marker algorithm or are just interested in extracting data from submitted programs. We were able to clarify that the underlying automatic marker architecture will not be changed in any way.

Fig 3.8 shows some of the snippets from the discussions we had during the initial focus group.

Student A: "It looks cool and nifty. I like some features like the average class score. For me the issue with the automatic marker is just how it presents the results. Imagine you had a system where the automatic marker has tabs with trial numbers, i.e., trial 1, trial 2, trial 3, which you could expand to see nicely formatted outputs for submitted programs in different colour. If the output is good, the trail tab goes green and if it is bad or incorrect, the trial tab goes red."

Student B: "Currently the diff utility is being used and the exact error codes from the tool are used. I was helping a student recently and asked them what does this error mean -2d2-. The student told me it is the diff utility and after we investigated, we noticed the error was because of a blank line being printed. Such small things make it hard to understand what the issue is with programs."

Student C: "Colour is important. If you use a good VCS diff tool, you will see that they have expected vs error of your program and this usually has different colours. Producing programs next to each other is even better. Currently, the output is shown below the program and you need to scroll a lot just to figure out what is wrong with your program."

Fig 3.8 Feedback from Participants during the Initial Focus Group Session

3.7. Iteration 1: Design and Feedback

After the initial discussions with the focus group, participants were asked to suggest features that they would like to see in the initial prototype design. To provide more context and help focus group participants answer the question, the following features (de Quincey et al., 2016) were shown to the students; however, they were told to add to this list if anything came to mind:

- Links to resources
- Time spent
- Social interaction/Communication
- Test results
- Emotion and stress analytics
- Accessibility options
- HCI/Design
- Student progress
- Scheduling
- Engagement scores
- Activity meters and Progress trackers
- Comparisons to colleagues
- Coursework hand-in dates and timetabling
- Errors generated/Popular errors

Based on the suggested features, the participants were able to suggest additional features that could be included in the design of the final mock-up. Fig 3.9 presents some of the responses received from the participants.

Student A: "This might be contentious, but a DPR tool might be useful where students can calculate what they need to make DP. Right now, we have a situation where students are trying to figure out what they need to make DP".

Student B: "The use of colours is something that we help a lot. People are more visual. Right now, you need to sift through a lot of text".

Student C: "A forum as a feature might idea. Although I think some courses already have Q and A tools in Vula".

Student D: "Error dictionary might be helpful. If you get certain kinds of errors, an error dictionary could show troubleshooting advice about errors".

Fig 3.9 Feedback from Participants on Dashboard Features

The students made various suggestions regarding features that they felt would be useful to have on an LA inspired dashboard used by the automatic marker (see Fig

3.9 above). At the end of the focus group session, students were asked to sketch an ideal LA dashboard based on all the discussions had during the focus group sessions.

3.8. Iteration 2: Design and Feedback

The sketches from Iteration 1 were used together with the feedback collected during focus group sessions to create a mock-up of the LA dashboard. The mock-ups were designed using Balsamiq[11]. Fig 3.10 and 3.11 below shows the second iteration of the dashboard mock-ups.

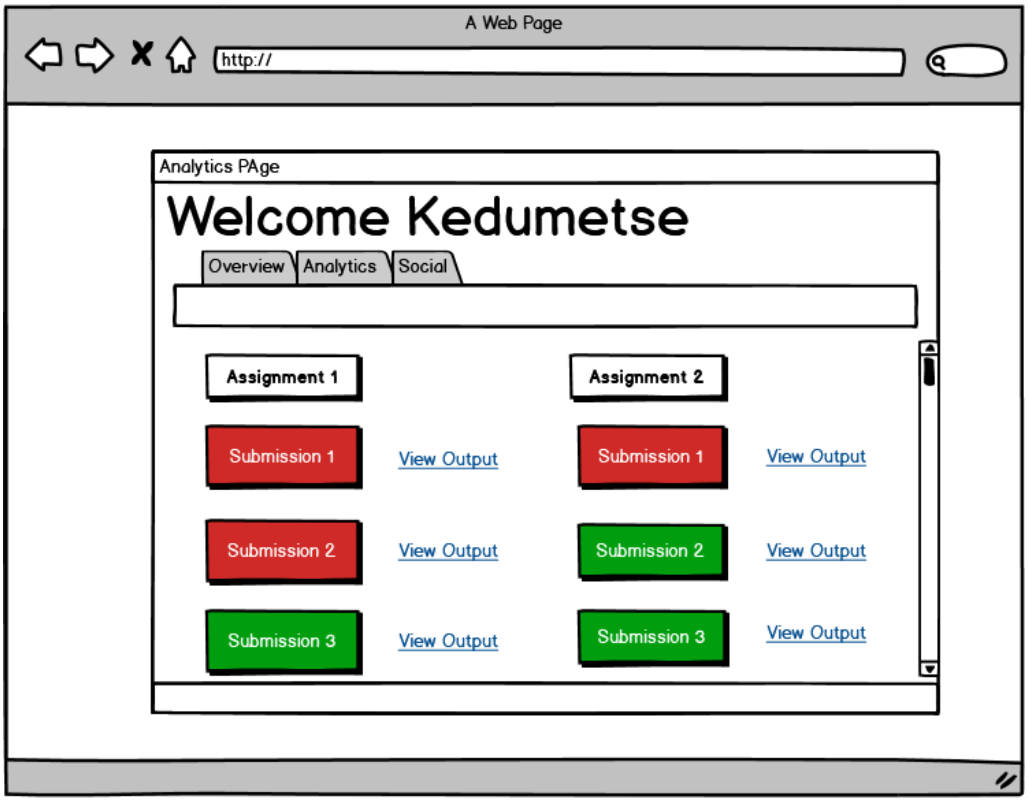


Fig 3.10 Initial Dashboard Mock-Up after Focus Group Interviews

The design shows a dashboard page, which has menu items, namely "Overview", "Analytics", and "Social". Only the design of the overview page was used in the initial mock-up. It showed two different columns, i.e., Assignment 1 and Assignment 2. Beneath these columns were menu buttons corresponding to the respective assignment submissions, i.e., "Submission 1", "Submission 2", and "Submission 3". The submission buttons were either green or red. Green represented a maximum score in the submission, while red represented a less than maximum score. Adjacent

[11] https://balsamiq.com/

to each of these buttons were links named “View output”. Students could click on these links and view the formatted output of their submitted programs. An example of the formatted output is presented in Figure 3.11 below.

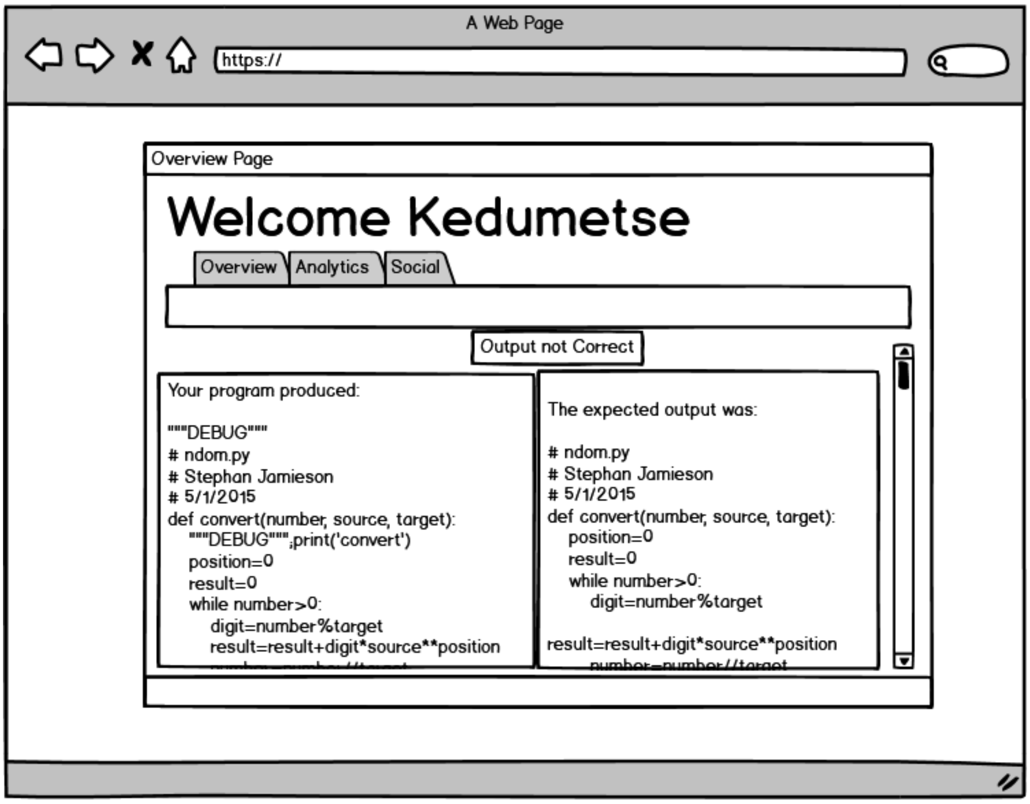


Fig 3.11 Formatted output from submitted programs

The mock-up depicted in Fig 3.11 shows an example of submitted program output with two columns. One column shows the output produced by the student’s submitted program, while the other column shows the expected output. Columns were placed adjacent to each other so that students could easily compare their output to the expected program output.

The feedback during the focus group interviews highlighted the need for an easy to use interface where the output from submitted programs could be easily interpreted by students. Some of the features that were suggested by the students, i.e., forums already existed within the UCT LMS, and there was thus no need to incorporate this as part of the dashboard design.

3.9. Iteration 3: Design and Feedback

After getting further feedback from the focus group participants about the designs of Fig 3.10 and 3.11, the mock-ups were iteratively improved. Fig 3.12 to 3.15 show the third iteration for the LA dashboard user interface. The main change suggested by the users was the need for the dashboard to display more information related to their overall performance on the "Overview Page". The redesign then, therefore, included more information such as "Highest Score", "Class Average", and "DP needed" to qualify for exams.

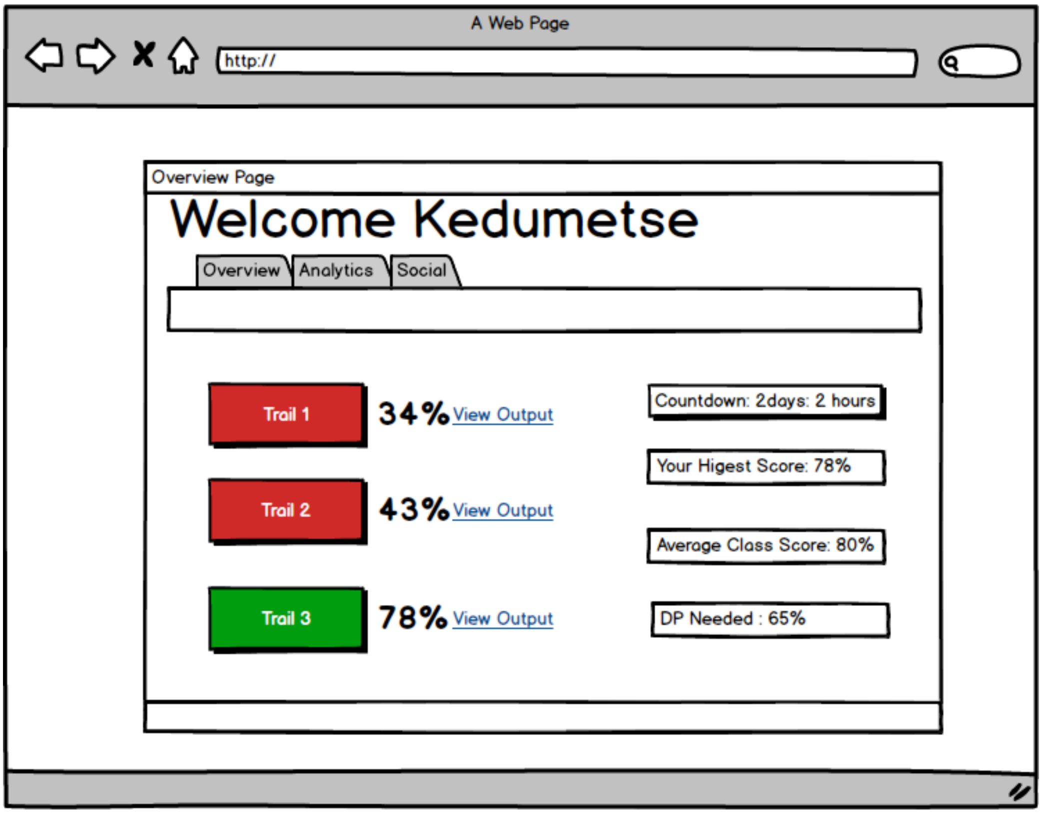


Fig 3.12 Overview Page of Dashboard Mock-Up

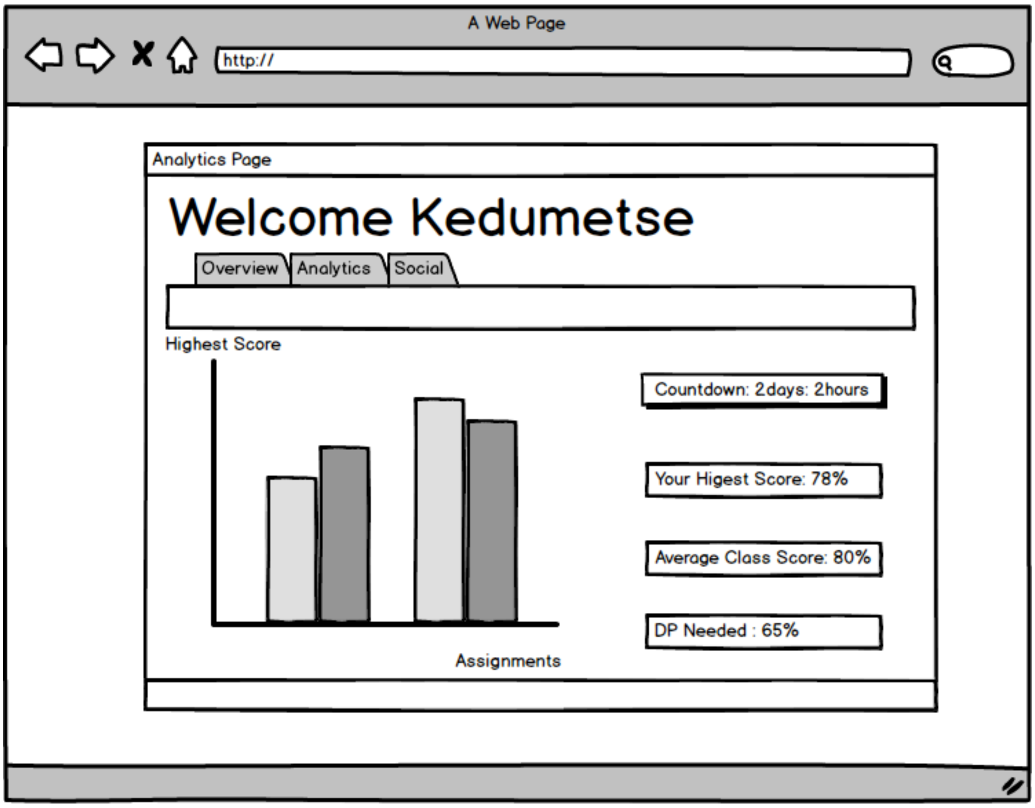


Fig 3.13 Analytics Page of Dashboard Mock-Up

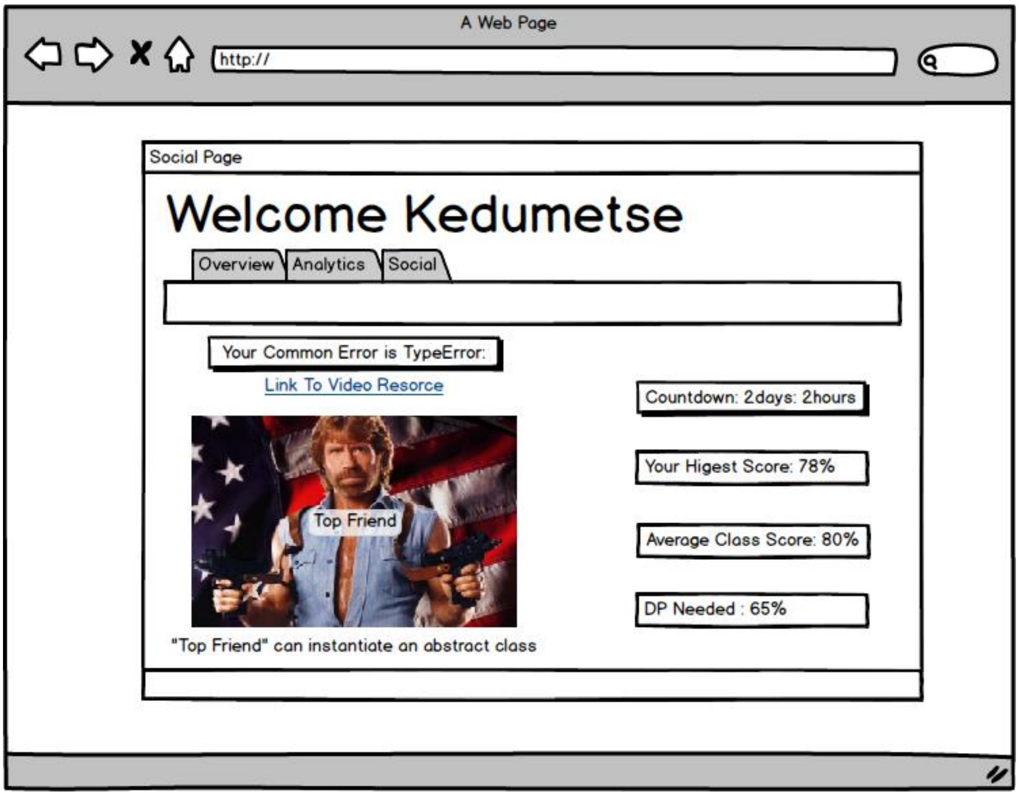


Fig 3.14 Social Page of Dashboard Mock-Up

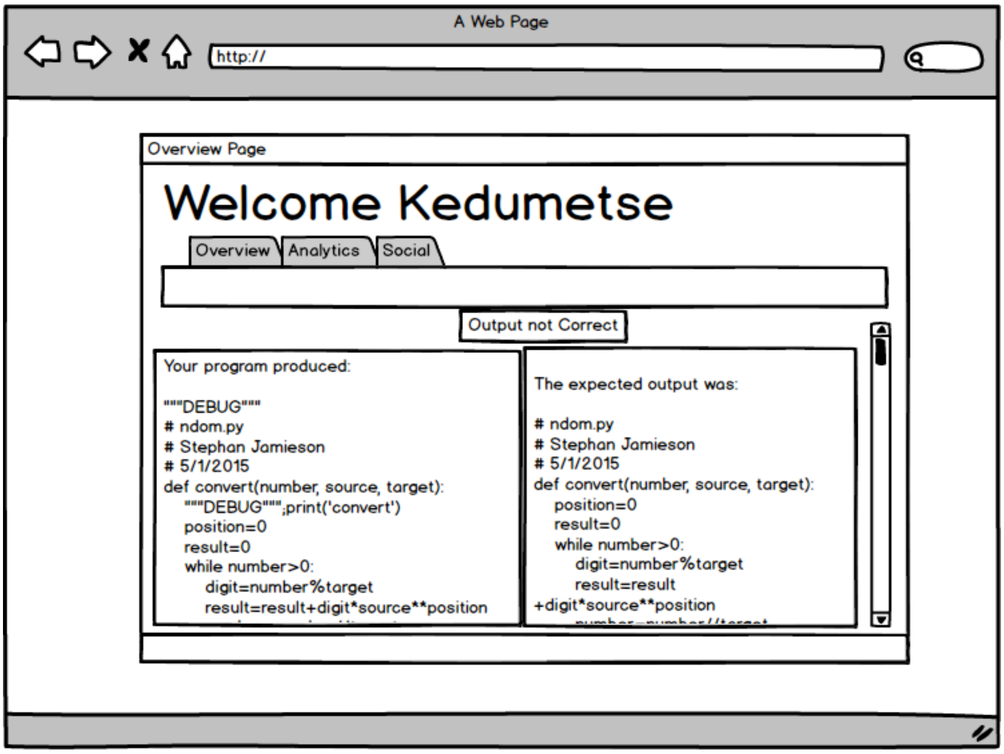


Fig 3.15 Program Output page of Dashboard Mock-Up

Based on the above mock-ups, a high-fidelity prototype dashboard (Fig 3.16 to 3.18) was developed using Python, specifically the Python library called Dash[12]. The code for the prototype is currently in Github[13].

Based on the final iteration, the final prototype of the LA dashboard interface was designed using the Plotly and Dash libraries and the Python programming language.

Fig 3.16 shows the overview page of the high-fidelity LA dashboard interface. The page displays metrics such as overall performance, when the next assignment is due, student's highest score, average class score, most common errors, and duly performed (DP) needed to qualify for exams.

[12] https://plot.ly/python/
[13] https://github.com/zambago/Learning-Analytics-Dashboard-

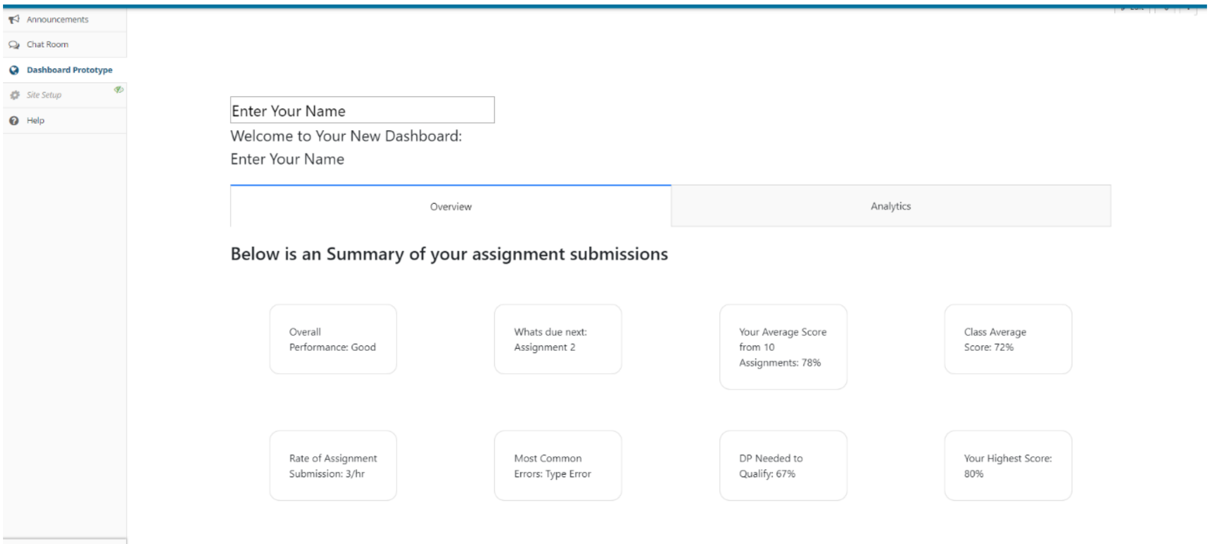


Fig 3.16 Overview Tab of Hosted Dashboard Embedded in Vula

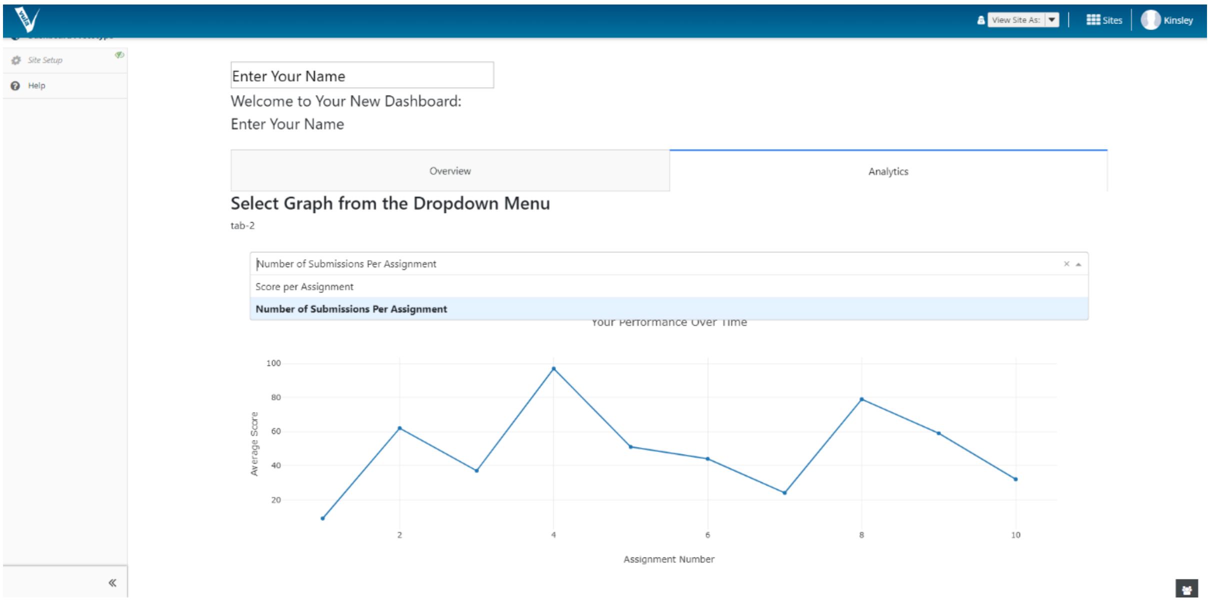


Fig 3.17 Analytics Tab Showing Student Progress Over Time

Fig 3.17 shows the analytics page and displays student performance over time base on all the assignment submissions. The high points on the graph show high assignment scores, while the low points show low assignment scores.

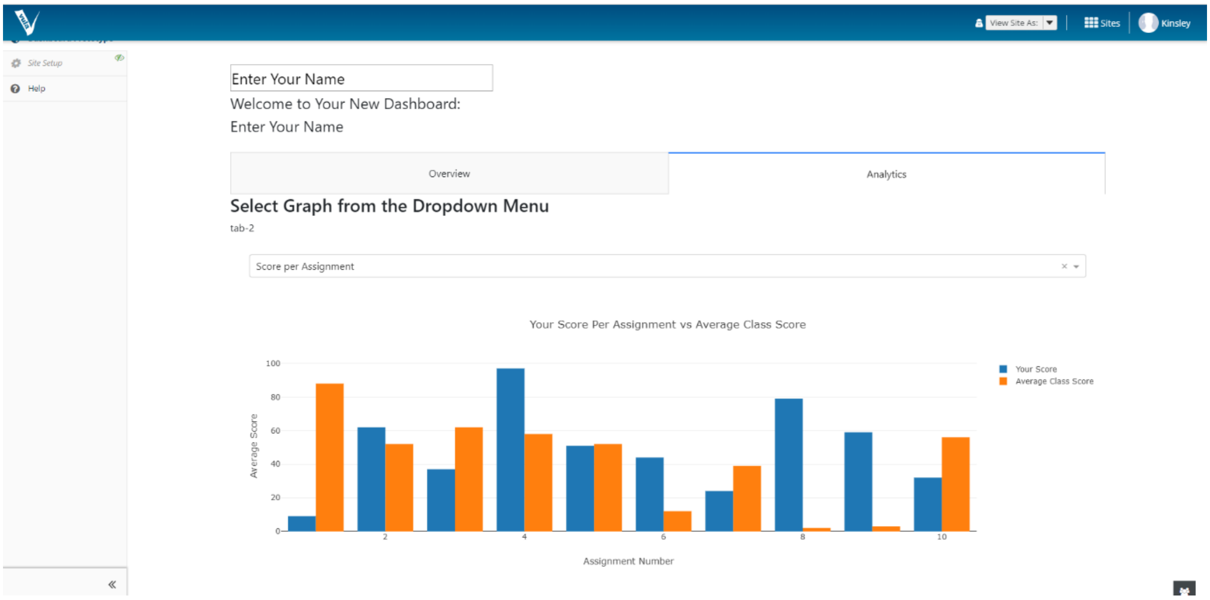


Fig 3.18 Analytics Tab Showing Student Scores Compared to Class Scores

Fig 3.18 shows another bar chart on the analytics page, showing student scores per assignment compared to the average class score for the same assignment. This provides an easy way for the student to compare their performance against that of their class.

3.10. Experiment Design

User feedback from both the paper prototype and first and second iterations were incorporated in the final design. To test the LA dashboard, an invitation was sent out to 30 first-year computer science students. Out of the 30 invites, 20 students confirmed their attendance and took part in the final usability study. Google forms was used to collect feedback using a survey. This study was conducted in the computer science lab at UCT, where students had access to computers to complete the questionnaires. A site was created in Vula[14] where 30 first-year computer science students were invited as participants in the final phase of the prototype evaluation. The dashboard prototype was hosted on Heroku and embedded as a webpage in the Vula site to simulate an experience of a real production LA inspired dashboard. Learners were then able to navigate the dashboard. After navigating through the prototype

[14] https://vula.uct.ac.za/portal

dashboard, students were required to complete a survey that was aimed at determining how useful or usable the dashboard was.

The final usability testing took place in the computer science laboratory between 11:00 am and 13:00 pm. A total of 20 students confirmed their participation in the usability experiment and were divided into two groups of 10. The first group was scheduled from 11:00 am until 12:00 noon, while the second group was scheduled from 12:00 noon until 13:00 pm. Students had access to the experiment Vula site, as shown in Fig 3.19.

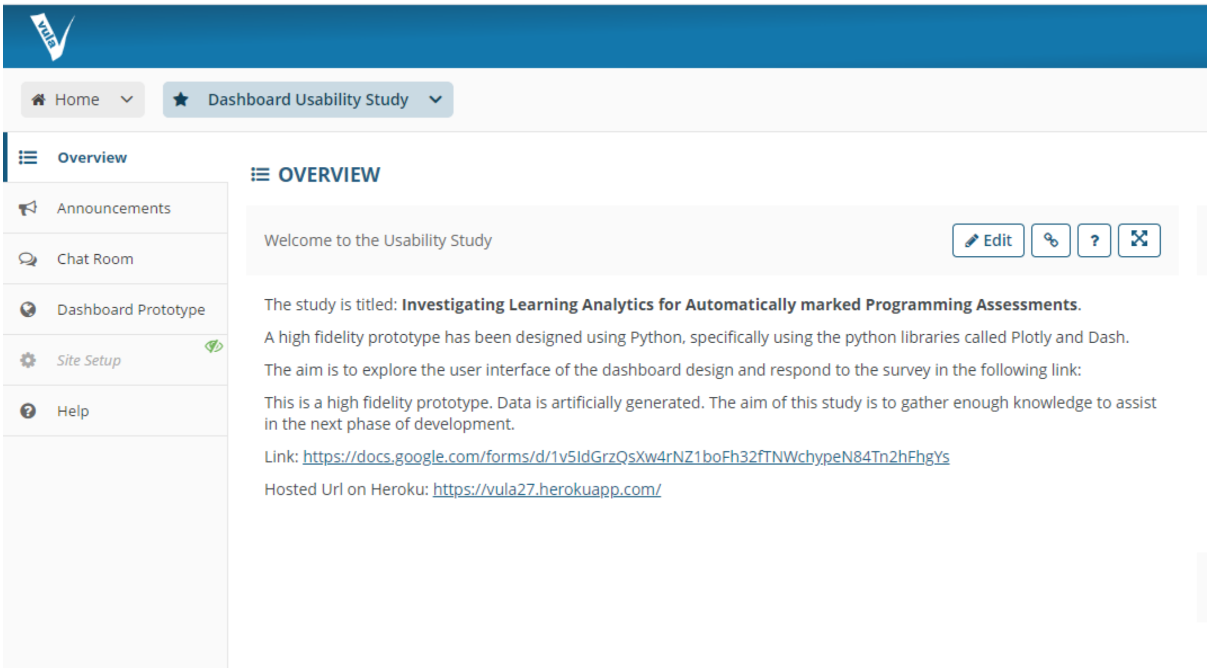


Fig 3.19 Overview Page of Dashboard Usability Study on Vula

Fig 3.19 shows the overview page in Vula. All the research participants were given access to this page a day before the experiment, allowing students to read the content and ask any questions before the experiment starts. The laboratory setting also ensured that students had access to computers and good Internet connectivity for the duration of the study. During the one-hour period, students were asked to log into the Vula site and visit the dashboard page. Students were instructed to click through the different pages of the dashboard. The overview page presented users with a metrics summary relating to assignment submissions. The data used for the high-fidelity dashboard prototype were generated by using the Python library, NumPy[15]. No actual data were generated from actual assignment submissions.

The analytics page presented users with two graphs, providing a summary of their assignment submissions over time and performance in relation to the average class performance. Once the users had gone through the LA dashboard interface and looked at the information presented on it, they were then required to complete two surveys – AttrakDiff[16] and Nielsen Usability Heuristics[17], which are discussed in detail in Chapter 4.

3.11. Chapter Summary

Throughout the study, different techniques were employed to collect data. The initial method involved focus groups with second-year computer science students and taking notes based on feedback received during the session. Students were presented with a rough dashboard sketch and guided by the interviewer by asking specific questions to understand student requirements for an LA dashboard. Feedback from these interviews was used in designing dashboard mock-ups. Feedback on the mock-up design was elicited from students in the Department of Computer Science at UCT and used to iteratively improve on the design of the prototype learning analytics dashboard. Based on these mock-ups, a high-fidelity prototype was designed using the Python programming language and other Python libraries. This prototype was then used to conduct a usability study with 20 first-year computer science students. Students were required to use the dashboard and respond to a questionnaire using Google forms. The feedback from this usability study was then analysed.

The next chapter (Chapter 4) presents the process of analysing the results of this study.

---

[15] https://numpy.org/
[16] http://AttrakDiff.de/index-en.html
[17] https://www.nngroup.com/articles/ten-usability-heuristics/

4. ANALYSIS AND RESULTS

This chapter gives an overview of how the results from the dashboard usability study were analysed. Two methods were used in the analysis, i.e., AttrakDiff (Hassenzahl et al., 2003) and Nielsen Usability Heuristics (Nielsen, 1994).

4.1. Analysis of Hedonic, Pragmatic Qualities and Attractiveness

AttrakDiff (Hassenzahl et al., 2003) was used as one of the core methods of investigation. AttrakDiff assesses how users personally rate the design of interactive products, i.e., measuring the usefulness of the product and its attractiveness by measuring hedonic qualities and pragmatic qualities. AttrakDiff was used because the researcher could assess the perceived user feelings about the system in the form of quantitative data, making use of the responses received (Phiri, 2018). The "Single Evaluation"[18] approach was used as opposed to the comparison A-B approach. The single evaluation approach was chosen because it provides a quick way to evaluate a single product by users. AttrakDiff has two other evaluation methods, namely c*omparison A-B* and *Before-After*. This could be useful for measuring variance in studies that employ more than one high-fidelity prototype and more than one group of participants.

In a case where there were two different dashboard prototypes, the comparison approach would have been better suited to help determine user preference for different designs.

The questionnaire consisted of three groups of questions, i.e., Pragmatic Quality (PQ), Hedonic Quality (HQ), and Attractiveness (ATT).

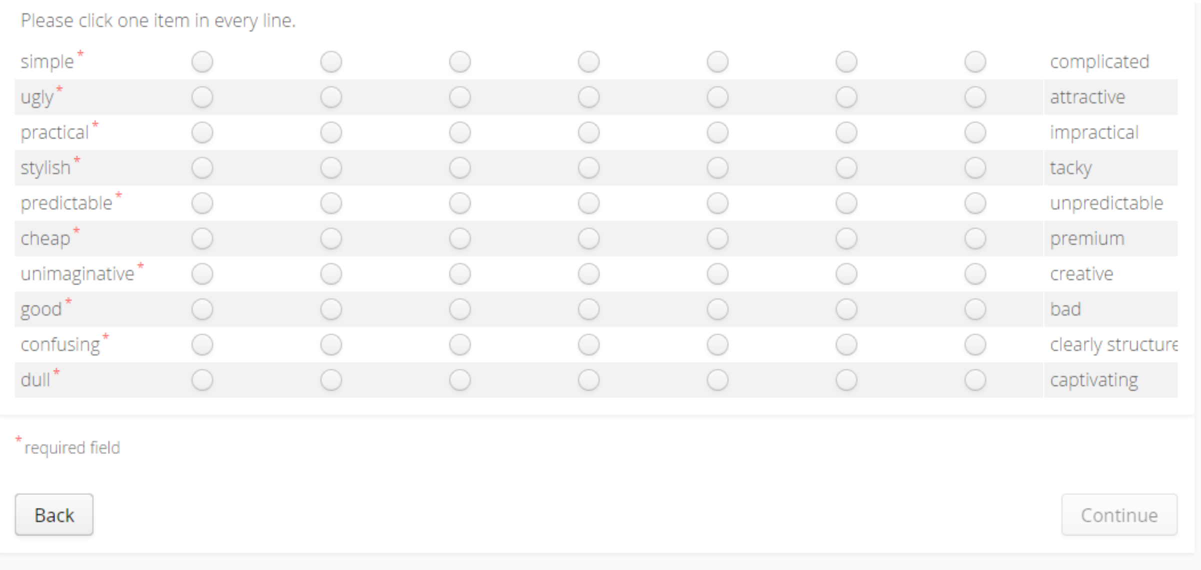
- PQ: Indicated the extent to which participants were successful at achieving the desired goal of a product, i.e., if the product functions were appropriate to achieving certain practical goals. This measures usability. PQs are task-oriented qualities.
- HQ: Indicated whether the product attributes could satisfy the human need to be perceived by others in a certain way. This measured emotional reactions.

[18] http://AttrakDiff.de/index-en.html

These were non-task-oriented qualities, i.e., whether it was presentable or stylish.

- ATT: Described the general positive or negative assessments of the product's appeal (Wetzlinger et al., 2014).

The instrument used in this study comprised of 10 opposite adjectives/word pairs rated on a 7-point Likert scale that specified the subjective contribution of each participant. Participants were asked to rate the dashboard application (see Fig 4.1). For each, the participant rated the application between two-word pairs.



Please click one item in every line.

simple *	○	○	○	○	○	○	○	complicated
ugly *	○	○	○	○	○	○	○	attractive
practical *	○	○	○	○	○	○	○	impractical
stylish *	○	○	○	○	○	○	○	tacky
predictable *	○	○	○	○	○	○	○	unpredictable
cheap *	○	○	○	○	○	○	○	premium
unimaginative *	○	○	○	○	○	○	○	creative
good *	○	○	○	○	○	○	○	bad
confusing *	○	○	○	○	○	○	○	clearly structure
dull *	○	○	○	○	○	○	○	captivating

* required field

Back

Continue

Fig 4.1 AttrakDiff Shortened Questionnaire

The word pairs were either simple or complicated, ugly or attractive, practical or impractical, stylish or tacky, predictable or unpredictable, cheap or premium, unimaginative or creative, good or bad, confusing or clearly structured, or dull or captivating. Selecting one (1) would indicate a strong preference to the words on the left while seven (7) would indicate a strong preference to the word/adjectives on the right.

From the responses received via the AttrakDiff survey, the portfolio diagram in Fig 4.2 was generated using the AttrakDiff console[19].

[19] https://esurvey.uid.com/project/#!login

Fig 4.2 Portfolio Presentation of Study

The vertical axis of Fig 4.2 has the HQ while the horizontal axis has the PQ. The bottom part of the vertical axis is the lowest value of HQ while the left side of the horizontal axis is the lowest value for PQ. Depending on the values of HQ and PQ from the survey, the product will lie in one of the “character-regions” in Fig 4.2. In this case, it lies in the task-oriented region. If the confidence rectangle is too big, it might become difficult to ascertain to which region it belongs. Small confidence rectangles are advantageous because it means the results of the investigation are reliable and less coincidental. If the users are aligned in their evaluation of the product from the survey feedback, the confidence rectangle will be small. If the evaluation ratings vary a lot, the confidence rectangle will be big.

In the case of our dashboard, the confidence rectangle is far to the right. This shows strong PQs, i.e., good ergonomics, perceived usability, and task-oriented quality. Examples of PQs include comprehensible, simple, predictable, and clear. The value of the PQ is 1.88 with a confidence of 0.28, while the HQ has a medium value of 0.64 with a confidence of 0.35. Examples of HQs responses include 'cheap' and 'unprofessional'. This is likely because this was only a prototype; hence the participants did not feel the dashboard was of a good standard. This could also have been caused by the fact that fake data were used to generate graphs for the prototype, and no real student data were used.

The average values of the AttrakDiff™ dimensions for the evaluated product are plotted in Fig 4.3.

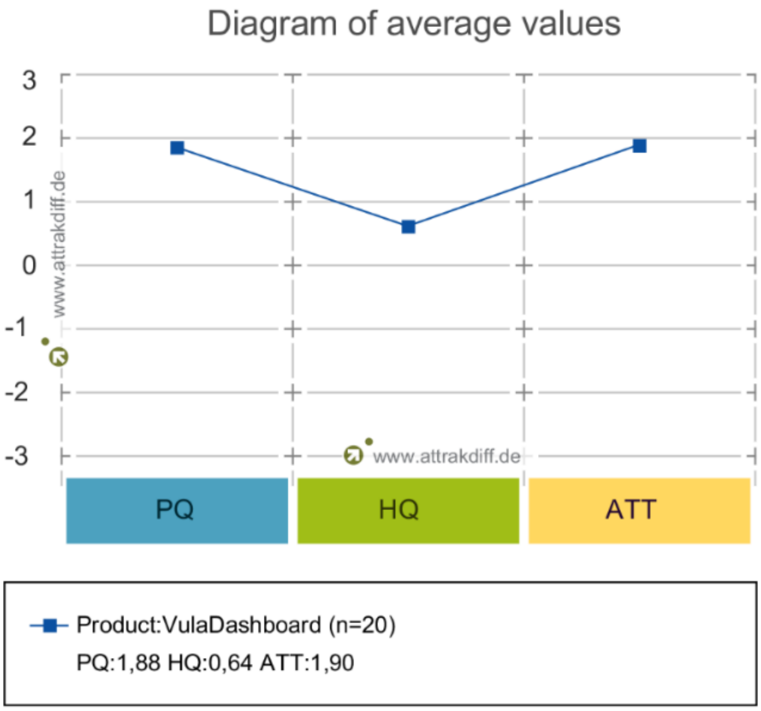


Fig 4.3 Diagram of Average Values

In this presentation, HQ distinguishes between the aspects of stimulation and identity. Furthermore, the rating of attractiveness is presented. This shows the users' rating of the attractiveness of the product. The diagram of average values shows a medium value for HQs (HQ = 0.64), while PQs (PQ = 1.88) and attractiveness (ATT = 1.90) have high values, which means that there was a strong positive perception regarding the usability of the LA inspired dashboard prototype.

Student data were finally analysed by plotting the participants' mean score ratings for each of the word pair categories in Fig 4.4.

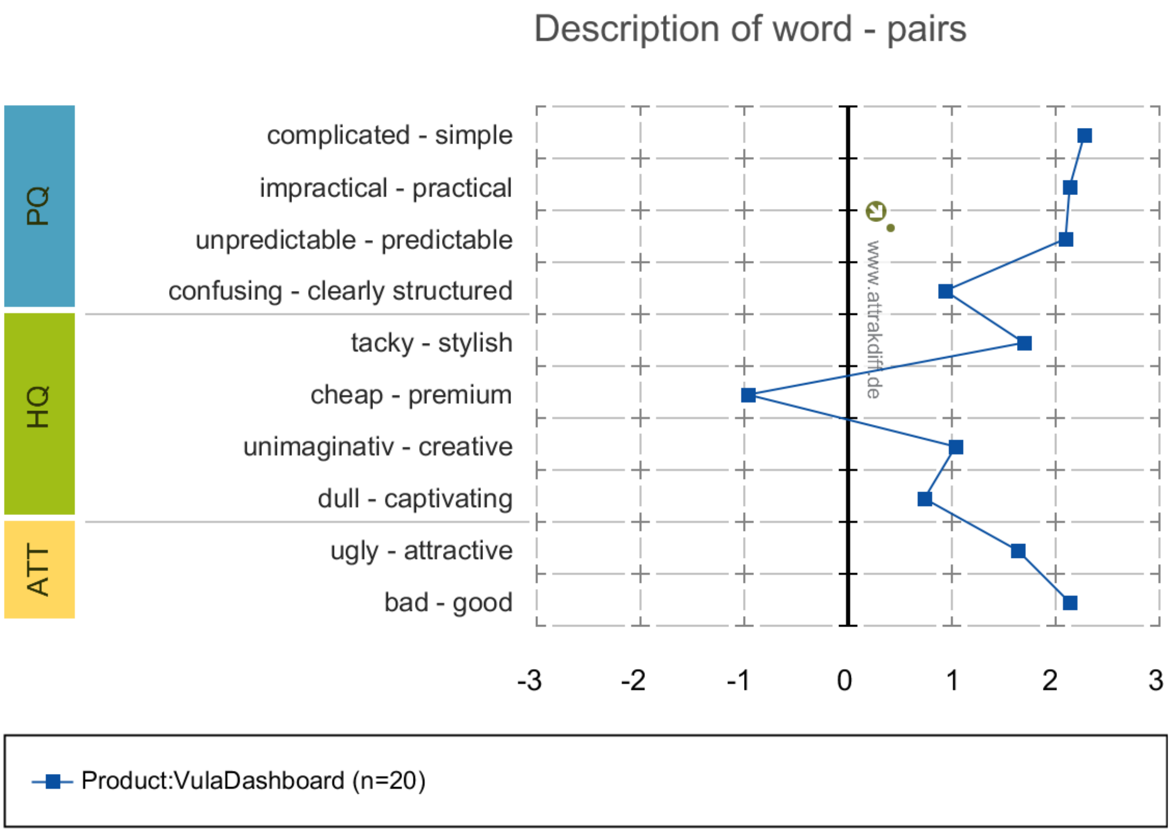


Fig 4.4 Description of Word Pairs

Mean values depict which characteristics are particularly critical or particularly well-resolved. From the word pair analysis, the "Premium to Cheap" word pair was of particular interest as the participants perceived the dashboard as cheap. Also, most participants viewed the dashboard as "simple" to use, "practical", "predictable", and "good".

4.2 Nielsen Usability Heuristics

Heuristic evaluation is a usability engineering method for evaluating user interfaces to find their usability problems (Nielsen, 1994). In this study, a set of evaluators, i.e., first-year computer science students, inspected the user interface of a high-fidelity prototype LA-inspired dashboard and rated the system based on the heuristics as proposed by Nielsen (1994). Table 4.1 shows the various heuristics, goals, and questions asked in the questionnaire for each heuristic.

Heuristic	Goal	Question asked in the questionnaire
Visibility of system status	The dashboard should be designed in such a way that users are informed of what is happening. They should receive the appropriate feedback in a reasonable timeframe.	• Does the dashboard have a header or "title" that describes its content? • Does the dashboard provide visuals that help you understand how to navigate and find information needed?
Match between the dashboard and real world	The dashboard should be designed with the user's context in mind, i.e., words, phrases, and concepts used in the design should be familiar to the user. Real-world conventions should be used, and information should appear in a natural and logical order.	• Is appropriate terminology used in the dashboard design? • Is the design of the dashboard aesthetically appealing?
User control and freedom	The dashboard should make it easy for the user to undo and redo actions. If the user clicks on a system function mistakenly, they should easily identify how to exit the unwanted state.	• Can you reverse actions easily when using the dashboard?
Consistency and standards	The word used in the design should not be ambiguous. User should not have to wonder about the meaning of different words, situations, or actions.	• Are the names of menu options consistent with every menu item of the dashboard in terms of grammar and terminology? • Is colour coding consistent throughout the dashboard?
Error prevention	The dashboard should be designed in a way that prevents errors from happening. Error-prone conditions should be eliminated. Users should be presented with a confirmation option before they commit an action which may be difficult to undo.	• Are the available menu choices logical and distinguishable in the dashboard design?
Recognition rather than recall	Objects, actions, and options should be visible to the user. This is intended at minimising the load on their memory. The user should not have to remember information from one part of the dialogue to another. Instructions for the use of the system should be visible or easily retrievable whenever appropriate.	• Are items grouped and placed logically and consistently throughout the dashboard? • Does the dashboard avoid the need to re-enter information already provided?
Flexibility and efficiency of use	The system should be efficient to use. Experienced users should be able to speed up their interaction with the system using accelerators. This might be unseen by the novice user. Users should be able to find information and tailor frequent	• Can you easily find highly desirable information?

Heuristic	Goal	Question asked in the questionnaire
	actions they perform on the system.	
Aesthetic and minimalist design	Dialogues should only contain relevant information that could help the user when using the system.	• Is the essential information to make decisions, and just that information, shown on the screen?
Help users recognise, diagnose, and recover from errors	Error messages should be easy to read. The problem should be precisely indicated, and a constructive solution suggested.	• Do the error messages announce the information using plain language?
Help and documentation	The system should be very easy to use, in a way that documentation is not needed. Although documentation is provided, it should be easy to search, focus on the task performed by the user, list concrete steps to be carried out and not be too large.	• Does the dashboard provide helpful information where needed?

Table 4. 1 Summary of Nielsen's Heuristics Used in this Study

Users were required to use the LA-inspired dashboard prototype and respond to the questions either with a "Yes", "No" or "Maybe".

The number of research participants for this usability study was:

$$N = 20$$

The following section discusses the usability heuristics responses in detail.

4.2.1. Visibility of System Status

The system should always keep users informed about what is going on through appropriate feedback within a reasonable time. To answer the question about the visibility of the dashboard status, the research participants were asked if the dashboard had a header and title and if the dashboard provided assistive feedback.

90% of the students said that the dashboard had a header and title that described its content, while 10% of the students responded that none was present.

90% of students agreed that the dashboard provided visuals that helped them navigate and find information. 5% said that there were no visuals, while the remaining 5% was not sure if the visuals helped them navigate or not.

4.2.2. Match between the System and Real World

The system should speak the users' language using words, phrases, and concepts familiar to the user rather than system-oriented terms, following real-world conventions, making information appear in a natural and logical order. To answer the question about the match between the system and the real world, the students were asked two questions.

In the first question, the students were asked if appropriate terminology was used in the dashboard design. 85% of the students agreed that the appropriate terminology had been used, while 5% said that was not the case. The remaining 10% was not certain if the appropriate terminology had been used.

The students were also asked if the design of the dashboard was aesthetically appealing. Most of the users agreed (55%) that the dashboard was aesthetically appealing, 30% responded with 'maybe' while 15% said it was not aesthetically appealing.

4.2.3. User Control and Freedom

Users often choose system functions by mistake and will, therefore, need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. The dashboard should make it easy for the user to undo and redo actions. If the user clicks on a system function mistakenly, they should easily identify how to exit the unwanted state. Students were asked if they could reverse actions in a simple way when using the dashboard.

40% of the students said they could reverse actions in a simple way, 30% were unsure, while 30% responded that they could not.

4.2.4. Consistency and Standards

Users should not have to wonder whether different words, situations, or actions mean the same thing. Students were asked if there was a consistent colour coding throughout the dashboard. 90% of the students agreed that the colour coding was consistent throughout the dashboard, while 10% responded that the colour coding was not consistent.

Students were also asked if the menu options of the dashboard had consistent names grammatically, and if the terminology used was consistent.

Majority of the students (80%) responded that the names of menu options are consistent with every menu item in the dashboard, whereas 15% said that these menu items were not consistent. 5% responded that maybe the menu items were consistent.

4.2.5. Error Prevention

Even better than good error messages is a careful design which prevents a problem from occurring in the first place. Students were asked if the available menu choices were logical and easy to distinguish in the dashboard design.

90% of the students responded that it was, while 10% said the menu options were not distinguishable.

4.2.6. Recognition rather than Recall

The user should not have to remember information from one part of the dialogue to another. Objects, actions, and options should be visible to the user. This is intended to minimise the load on their memory. Instructions for the use of the system should be visible or easily retrievable whenever appropriate. To answer this question, students were asked two questions. The first was meant to find out whether items were grouped and placed logically throughout the dashboard. Secondly, students were asked whether the dashboard design made it possible for them to avoid entering information already provided.

Most of the respondents (85%) said that the items in the dashboard design were logically placed, 5% disagreed while 10% were not sure whether the items were placed logically.

75% of the students responded that the dashboard did not require them to re-enter information already provided. In comparison, 10% responded that it did not eliminate the need to re-enter information that was already provided. The remaining 15% were not sure whether the need to re-enter already provided information had been eliminated.

4.2.7. Flexibility and Efficiency of Use

The system should be efficient to use. Experienced users should be able to speed up their interaction with the system using accelerators, which might be unseen by the novice user. Users should be able to find information and tailor frequent actions they perform on the system. Students were asked whether they could easily find desirable information on the dashboard.

Most of the students (95%) said that they could easily find desirable information on the dashboard, while 5% said they could not.

4.2.8. Aesthetic and Minimalist Design

Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility. Students were asked whether the dashboard user interface showed only the essential information needed to help them make decisions.

Most students (80%) responded that just the essential information required to make decisions was shown on the dashboard, 10% responded that this was not the case while 10% were not sure whether only essential information was shown on the dashboard.

4.2.9. Help Users Recognise, Diagnose, and maybe Recover from Errors

Error messages should be expressed in plain language. The problem should be precisely indicated, and a constructive solution suggested. Students were asked whether the error messages on the dashboard used plain language and were easy to understand.

Most of the students (50%) were not sure if the error messaging in the dashboard design was appropriate. 40% said that the error messaging helped them to diagnose problems, while 10% said that it did not.

4.2.10. Help and Documentation

The system should be very easy to use, in a way that documentation is not needed. Although documentation is provided, it should be easy to search, focus on the task performed by the user, list concrete steps to be carried out, and not be too large. Students were asked whether the dashboard presented helpful information to them when it was needed.

60% of the students responded that the dashboard provided helpful information where needed, while 10% said that no information was provided. The remaining 30% were not sure.

4.3. Chapter Summary

This chapter shows how two user experience methods were used, namely AttrakDiff and Nielsen's Usability Heuristics in the analysis of an LA inspired dashboard. The results of the study are presented in the form of visualisation that shows responses of either "Yes", "No" or "Maybe" to specific questions that were asked through a survey. The responses from all Nielsen's heuristics are presented in Fig 4.5.

Using the average from all the responses received from the study participants, 77% of the respondents said that the dashboard was usable, while 11% responded that it was not usable. Furthermore, 14% of the respondents were not sure about its usability and this could have been caused by issues related specifically to the design of the dashboard prototype, such as the time and resource constraints to enhance the protype.

95% indicated that they could easily find the desired information within the dashboard. 90% of students said that the available menu options were logical, the colour coding was consistent, and the dashboard provided visuals that helped them navigate and find the information they needed.

AttrakDiff measured the PQ, HQ and ATT of the user experience. The aim of measuring the pragmatic qualities was to find out to what extent the participants were successful in achieving practical goals or task-oriented activities. HQs measured emotional reactions or non-task-oriented activities, i.e., how stylish or presentable the students felt the dashboard was. The analysis shows high values of PQ. This meant that the students felt that the dashboard used for this study enabled them to achieve their practical goals. The ATT value was 1.90. ATT measured the negative or positive assessments of the product's appeal such as ugly or attractive and good or bad. The positive value of 1.90 shows the product was positively appealing to the users.

The results for the visual appearance of the dashboard were mixed. Thirty percent (30%) of students responded that the dashboard was not aesthetically appealing. In response to how well the dashboard provides feedback on error messages, 40% of the students said that it did not provide enough feedback. Furthermore, 30% of students responded that the dashboard did not provide helpful information where needed. The value of HQ was 0.66 which was medium. HQ indicates whether the product attributes satisfy the human needs. It measured emotional reactions and non-task-oriented activities such as the presentation of the dashboard. Most of the users responded that the dashboard was cheap and unimaginative. This rather low value of HQ was expected because this was only a prototype and a lot of effort was not put into aesthetics. Although there was a high ATT because users responded that the dashboard was good in terms of fulfilling their needs, the value of HQ relays their feelings towards the overall quality of the prototype dashboard design.

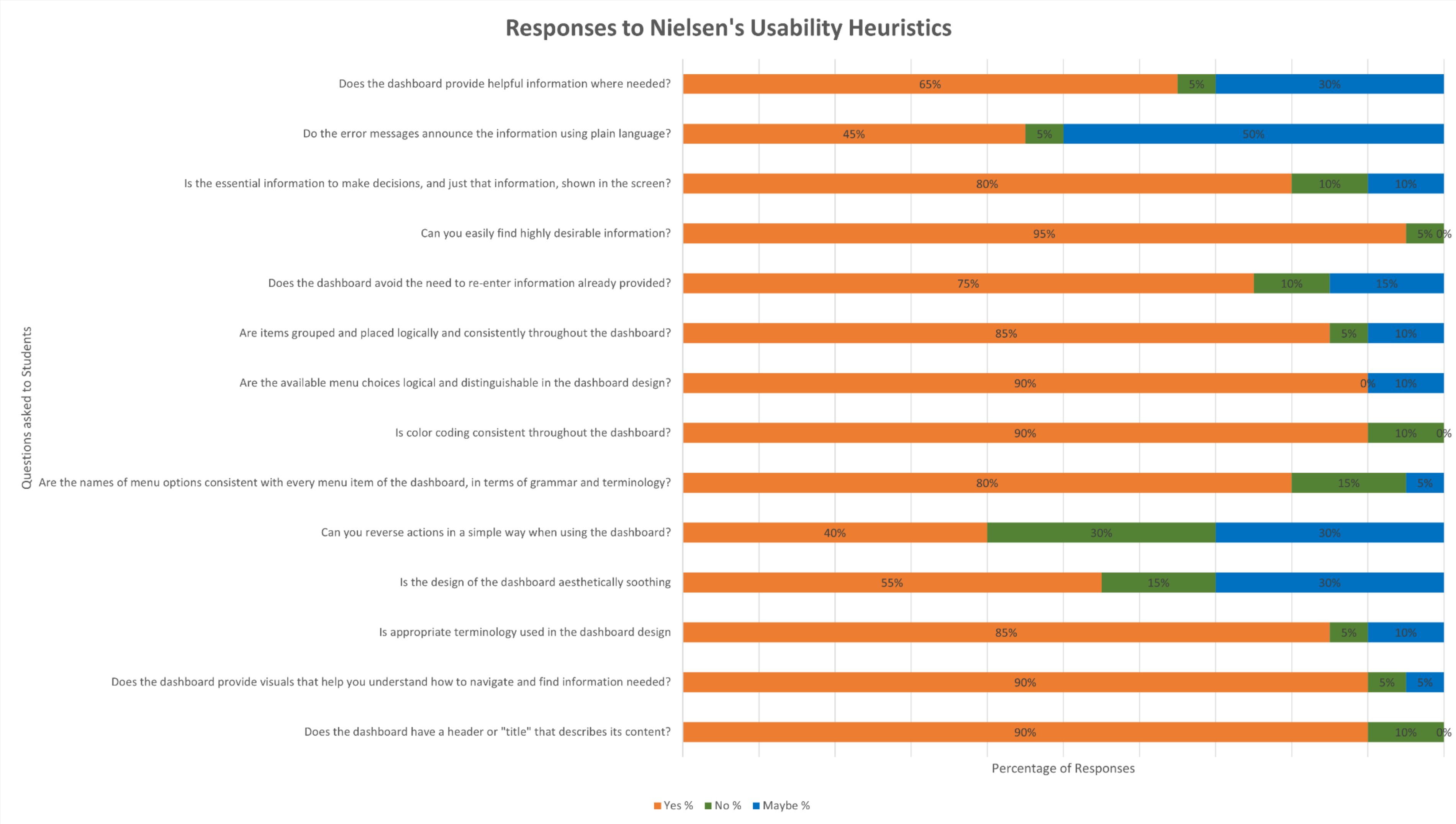


Fig 4.5 Responses and their respective percentages

5. CONCLUSION, LIMITATIONS AND FUTURE RESEARCH

5.1. Conclusion

The main aim of the study was to investigate an LA interface for automatically marked programming assessments.

The research question posed was: “*Can an LA dashboard provide students with useful feedback in automatically marked programming assessments?”*

To answer the research question, a prototype LA interface was designed and tested with first year Computer Science students. User-centred methods were employed in the research and design of the interface. Nielsen’s usability heuristics and Attrakdiff were used to analyse responses of students who used this prototype. The user-centred design methodology was used because it allows researchers to understand users deeply and creates the opportunity to implement designs that are needed by the communities they design for. This group of students was chosen because they already use Vula for the submission of programming assignments that are marked automatically. Therefore, they could investigate the usability and usefulness of an LA dashboard, which visualises submitted programming assessments through the Vula LMS used at the UCT.

Data collection was done through focus group activities, which included interviews and usability surveys. Based on the feedback from the focus group, several iterations of low-fidelity prototype dashboards were designed. From these low-fidelity prototypes, a high-fidelity prototype LA dashboard was designed and tested with Computer Science students from the UCT.

The students were issued with Nielsen’s usability heuristic survey and AttrakDiff to evaluate the LA interface prototype. AttrakDiff allowed us to measure hedonic qualities and pragmatic qualities in the prototype dashboard design. PQs measured the practical uses of the dashboard, i.e., how useful the dashboard design was for a particular task the students wanted to complete. HQs measured the emotional qualities, such as the attractiveness of the dashboard design. Nielsen’s Usability

Heuristics evaluated the usability problems that students encountered while using the prototype. There was a high PQ, which meant that the dashboard interface was appropriate for helping the students achieve certain goals. 77% of the students responded that the dashboard was usable and useful to them based on Nielsen's heuristics. There was a strong positive perception regarding the usability of the LA dashboard interface.

In response to the research question that encapsulated this study, the LA dashboard prototype provided useful feedback for programming assessments to Computer Science students. Although students have always received feedback from automatically marked programming assignments, most of them complained that the feedback was difficult to understand and interpret. To address this problem, the prototype dashboard design presented information in a more structured way. User-centred design helped us to understand what information was useful to the students and the best way to present it on the dashboard interface.

5.2. Limitations and Future Work

This study involved only first- and second-year Computer Science students. However, the automatic marking system is used by students across different undergraduate and postgraduate levels and students in other departments in the Science faculty. A cross-functional study involving these different categories of students could result in a more representative design.

The prototype was tested on a single group of research participants, namely first-year students. Also, only one high-fidelity prototype design was tested. Future in-depth studies can be conducted where more than one high fidelity prototype is tested with more than one group of participants.

A single evaluation AttrakDiff method was used for data collection. AttrakDiff has two other evaluation methods, namely c*omparison A-B* and *Before-After*. This could be useful for measuring variance in studies that employ more than one high-fidelity prototype and more than one group of participants.

Future areas of research involve implementing the LA dashboard prototype designed in this study in a live learning environment where students can view submissions of programming assignments and provide feedback to improve the prototype. Closely related to this will be researching the use of predictive analytics to identify at-risk first-year Computer Science students who are learning to program based on the kinds of errors received from submitted programs.

Bibliography

Aleven, V., Xhakaj, F., Bodily, R., Kay, J., Davis, D., Jivet, I., & Verbert, K. (2018). *Open learner models and learning analytics dashboards*. March 7–9, 2018, Sydney, NSW, Australia, 41–50. https://doi.org/10.1145/3170358.3170409

Ali, L., Hatala, M., Gašević, D., & Jovanović, J. (2012). A qualitative evaluation of evolution of a learning analytics tool. *Computers & Education*, *58*(1), 470–489. https://doi.org/10.1016/j.compedu.2011.08.030

Alikaniotis, D., Yannakoudakis, H., & Rei, M. (2016). Automatic Text Scoring Using Neural Networks. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers),* Berlin, Germany, 2016, 715–725. https://doi.org/10.18653/v1/P16-1068

Blumenstein, M., Green, S., Nguyen, A., & Muthukkumarasamy, V. (2004). GAME: A generic automated marking environment for programming assessment. *International Conference on Information Technology: Coding Computing, ITCC*, April 5-7, 2004, Las Vegas, Nevada, USA, *1*, 212–216. https://doi.org/10.1109/ITCC.2004.1286454

Bolton, G. (2010). *Reflective practice writing for professional development* (3rd edition). Sage Publications.

Chatti, M. A., Dyckhoff, A. L., Schroeder, U., & Thüs, H. (2012). A reference model for learning analytics. *International Journal of Technology Enhanced Learning*, *4*(5/6), 318–240. https://doi.org/10.1504/IJTEL.2012.051815

Clow, D. (2012). The learning analytics cycle: Closing the loop effectively. *Proceedings of the 2nd International Conference on Learning Analytics and Knowledge – LAK '12*, 134. https://doi.org/10.1145/2330601.2330636

Corrin, L., & de Barba, P. (2015). How do students interpret feedback delivered via dashboards? *Proceedings of the Fifth International Conference on Learning Analytics and Knowledge – LAK '15*, Poughkeepsie, New York, 430–431. https://doi.org/10.1145/2723576.2723662

Dalgarno, B., Chan, A., Adams, P., Roy, P., & Miller, D. (2007). On campus and distance student attitudes towards paperless assessment and feedback. *Ascilite*, Singapore, 2 – 5 December 2007, 168–178.

De Freitas, S., Irving, L., Verbert, K., Gibson, D., Star, K., Alvarez, V., & Charleer, S. (2019). How to use gamified dashboards and learning analytics for providing immediate student feedback and performance tracking in higher education. *26th International World Wide Web Conference 2017, WWW 2017 Companion*, Perth Australia, *May 2018*, 429–434. https://doi.org/10.1145/3041021.3054175

De Quincey, E., Turner, M., Williams, N., & Kyriacou, T. (2016). Learner Analytics; The Need for User-Centred Design in Learning Analytics. *ICST Transactions on Ambient Systems*, *3*(9), 151643. https://doi.org/10.4108/eai.23-8-2016.151643

Deakin Crick, R., Huang, S., Ahmed Shafi, A., & Goldspink, C. (2015). Developing Resilient Agency in Learning: The Internal Structure of Learning Power. *British Journal of Educational Studies*, *63*(2), 121–160. https://doi.org/10.1080/00071005.2015.1006574

Díaz, H. J. P., Kloos, C. D., Ruiz, J. S., Ruipérez-Valiente, J. A., & Muñoz-Merino, P. J. (2014). *Towards the development of a learning analytics extension in open edX*, 299–306. https://doi.org/10.1145/2669711.2669914

Dourish, P., & Bellotti, V. (1992). Awareness and coordination in shared workspaces. *Proceedings of the 1992 ACM Conference on Computer-Supported Cooperative Work – CSCW '92*, Ontario,Canada, 107–114. https://doi.org/10.1145/143457.143468

Durall, E., & Analysis, N. (2014). *Learning Analytics as Metacognitive Tool*. Barcelona, Spain, 1 -3 April 2014, 380–384. https://doi.org/10.5220/0004933203800384

Echeverria, V., Martinez-Maldonado, R., Granda, R., Chiluiza, K., Conati, C., & Shum, S. B. (2018). Driving data storytelling from learning design. *Proceedings of the 8th International Conference on Learning Analytics and Knowledge*, Sydney Australia, March 5-9, 131–140. https://doi.org/10.1145/3170358.3170380

Echeverria, V., Martinez-Maldonado, R., Shum, S. B., Chiluiza, K., Granda, R., & Conati, C. (2018). Exploratory versus Explanatory Visual Learning Analytics: Driving Teachers' Attention through Educational Data Storytelling. *Journal of Learning Analytics*, *5*(3), 72–97. https://doi.org/10.18608/jla.2018.53.6

Faniran, V., & Ajayi, N. (2016). Students' perceptions of computer-based assessments: A case of UKZN. *2016 IST-Africa Conference, IST-Africa 2016*, Durban, South Africa, May 11-13, 1–9. https://doi.org/10.1109/ISTAFRICA.2016.7530692

Finlay, J., Connor, A., & Pears, R. (2011). Mining Software Metrics from Jazz. *2011 Ninth International Conference on Software Engineering Research, Management and Applications*, Zurich, Switzerland, November 21 -22, 39–45. https://doi.org/10.1109/SERA.2011.40

Froese, M. E., & Tory, M. (2016). Lessons Learned from Designing Visualization Dashboards. *IEEE Computer Graphics and Applications*, *36*(2), 83–89. https://doi.org/10.1109/MCG.2016.33

Fu, X., Shimada, A., Ogata, H., Taniguchi, Y., & Suehiro, D. (2017). Real-time learning analytics for C programming language courses. *Proceedings of the Seventh International Learning Analytics & Knowledge Conference on LAK '17*, Vancouver, Canada, March 13-17, 280–288. https://doi.org/10.1145/3027385.3027407

Fukuzumi, S., Noda, N., & Tanikawa, Y. (2017). How to Apply Human-Centered Design Process (HCDP) to Software Development Process? *Proceedings - 2017 IEEE/ACM 1st International Workshop on Design and Innovation in Software Engineering, DISE 2017*, Buenos Aires, May 23, 13–16, 13–16. https://doi.org/10.1109/DISE.2017.10

Gruzd, A., & Conroy, N. (2018). Designing a learning analytics dashboard for twitter-facilitated teaching. *Proceedings of the Fifth Annual ACM Conference on Learning at Scale – L@S '18*, London, United Kingdom, June 2018, 1–4. https://doi.org/10.1145/3231644.3231704

Hassenzahl, M., Burmester, M., & Koller, F. (2003). AttrakDiff: Ein Fragebogen zur Messung wahrgenommener hedonischer und pragmatischer Qualität. In G. Szwillus & J. Ziegler (Eds.). *Mensch & Computer, 57*, 187–196. Vieweg+Teubner Verlag. https://doi.org/10.1007/978-3-322-80058-9_19

Heaney, D., & Daly, C. (2004). Mass production of individual feedback. *ACM SIGCSE Bulletin*, *36*(3), 117. https://doi.org/10.1145/1026487.1008029

Ivanov, V., Pischulin, V., Rogers, A., Succi, G., Yi, J., & Zorin, V. (2018). Design and validation of precooked developer dashboards. *Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering – ESEC/FSE 2018*, Lake Buena Vista, FL USA, November 2018, 821–826. https://doi.org/10.1145/1026487.1008029

Jivet, I., Scheffel, M., Specht, M., & Drachsler, H. (2018). License to evaluate. *Proceedings of the 8th International Conference on Learning Analytics and Knowledge – LAK '18*, 31–40. https://doi.org/10.1145/3170358.3170421

Kitto, K., Lupton, M., Davis, K., & Waters, Z. (2016). Incorporating student-facing learning analytics into pedagogical practice. *Proceedings of the Annual ASCILITE Conference*, Adelaide, Australia, November 28 -30*,* 1–10. https://doi.org/10.1517/17425255.4.2.193

Kitto, K., Shum, S. B., & Gibson, A. (2018). Embracing imperfection in learning analytics. *Proceedings of the 8th International Conference on Learning Analytics and Knowledge – LAK '18*, Sydney, Australia, March 2018, 451–460. https://doi.org/10.1145/3170358.3170413

Li, L., Li Xiuhua, & Wang Qiulan. (2010). A flexible computer-based assessment system that uses flash disk. *2010 International Conference on Educational and Network Technology*, Qinhuangdao, China, June 25-27, 426–429. https://doi.org/10.1109/ICENT.2010.5532270

MacWilliam, T., & Malan, D. J. (2013). Streamlining Grading toward Better Feedback. *ITiCSE'13, Canterbury, England, UK*, *July 1–3, 2013,* 147–152. https://doi.org/10.1145/2462476.2462506

Mansouri, F. Z., Gibbon, C. A., & Higgins, C. A. (1998). PRAM: PRolog Automatic Marker. *ITiCSE '98*, Dublin, Ireland, August 18-28, 166–170. https://doi.org/10.1145/282991.283108

Mothukuri, U. K., Reddy, B. V., Reddy, P. N., Gutti, S., Mandula, K., Parupalli, R., Murty, C. H. A. S., & Magesh, E. (2017). Improvisation of learning experience using learning analytics in eLearning. *2017 5th National Conference on E-Learning & E-Learning Technologies (ELELTECH)*, Hyderabad, India, August 3 -4, 1–6. https://doi.org/10.1109/ELELTECH.2017.8074995

Ngo-Pham, K., Bigham, J., Rodaway, J., & Tokarchuk, L. (2005). A system to mark programs automatically. *Proceedings – International Conference on Quality Software 2005*, Victoria, Australia, September 19-20, 317–322. https://doi.org/10.1109/QSIC.2005.9

Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. *Conference on Human Factors in Computing Systems Proceedings*, Boston Massacusetts, April 24-28,152–158. https://doi.org/10.1145/259963.260333

Olivares, D. (2015). Exploring Learning Analytics for Computing Education. *Proceedings of the Eleventh Annual International Conference on International Computing Education Research – ICER '15*, Omaha, Nebraska, August 9-13, 271–272. https://doi.org/10.1145/2787622.2787746

Phiri, L. (2018). *Investigating the Impact of Organised Technology-driven Orchestration on Teaching* (Doctoral dissertation, University of Cape Town, Cape Town, South Africa).

Picciano, A. G. (2012). The Evolution of Big Data and Learning Analytics in American Higher Education. *Online Learning*, *16*(3), 9–20. https://doi.org/10.24059/olj.v16i3.267

Pieterse, V. (2013). Automated Assessment of Programming Assignments. *Computer Science Education Research Conference (CSERC'13)*, Arnhem, Netherlands, April 4-5, *1* (April), 12.

Roberts, L. D., Howell, J. A., & Seaman, K. (2017). Give Me a Customizable Dashboard: Personalized Learning Analytics Dashboards in Higher Education. *Technology, Knowledge and Learning*, *22*(3), 317–333. https://doi.org/10.1007/s10758-017-9316-1

Roberts, L. D., Howell, J. A., Seaman, K., & Gibson, D. C. (2016). Student attitudes toward learning analytics in higher education: 'The fitbit version of the learning world'. *Frontiers in Psychology*, *7*(DEC), 1–11. https://doi.org/10.3389/fpsyg.2016.01959

Sanna, A., Lamberti, F., Paravati, G., & Demartini, C. (2012). Automatic Assessment of 3D Modeling Exams. *IEEE Transactions on Learning Technologies*, *5*(1), 2–10. https://doi.org/10.1109/TLT.2011.4

Scheffel, M., Drachsler, H., Stoyanov, S., & Specht, M. (2014). Quality Indicators for Learning and Analytics. *Educational Technology & Society*, *17*(July), 117–132. https://doi.org/10.1002/chem.201202900

Schwendimann, B. A., Boroujeni, M. S., Holzer, A., Gillet, D., & Dillenbourg, P. (2015). Understanding learning at a glance: An overview of learning dashboard studies Categories and Subject Descriptors. *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge*, Edinburgh, UK, April 25-29, 532–533. https://doi.org/10.1039/c0ee00518e

Schwendimann, B. A., Rodríguez-Triana, M. J., Vozniuk, A., Prieto, L. P., Boroujeni, M. S., Holzer, A., Gillet, D., & Dillenbourg, P. (2016). Understanding learning at a glance. *Proceedings of the Sixth International Conference on Learning Analytics & Knowledge – LAK '16*, Edinburgh, UK, April 25-29, 532–533. https://doi.org/10.1145/2883851.2883930

Seffah, A., Metzker, E.: The obstacles and myths of usability and software engineering. Commun. ACM 47(12), 71–76 (2004)

Suleman, H. (2008). Automatic marking with Sakai. *Proceedings of the 2008 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries Riding the Wave of Technology – SAICSIT '08*, Wilderness, South Africa, Cotober, 2008, 229–236. https://doi.org/10.1145/1456659.1456686

Thomas, P. (2003). The evaluation of electronic marking of examinations. *Self*, 50–54. https://doi.org/10.1145/961511.961528

Thomas, P., Waugh, K., & Smith, N. (2005). *Experiments in the Automatic Marking of ER-Diagrams*. *ITiCSE 05*, Monte de Caparica, portugal, june 27-29, 6.

Treude, C., & Storey, M.-A. (2010). Awareness 2.0: Staying aware of projects, developers and tasks using dashboards and feeds. *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering – ICSE '10*, Cape Town, South Africa, May 2010, *1*, 365. https://doi.org/10.1145/1806799.1806854

Verbert, K., Govaerts, S., Duval, E., Santos, J. L., Van Assche, F., Parra, G., & Klerkx, J. (2014). Learning dashboards: An overview and future research opportunities. *Personal and Ubiquitous Computing*, *18*(6), 1499–1514. https://doi.org/10.1007/s00779-013-0751-2

Walsh, T., Kujala, S., Petrie, H., & Power, C. (2014). *Axe UX: Exploring Long-Term User Experience with iScale and AttrakDiff*. AcademicMindTrek, Tampere, Finland, November 4-7, 32–39.

Wetzlinger, W., Auinger, A., & Dörflinger, M. (2014). Comparing Effectiveness, Efficiency, Ease of Use, Usability and User Experience When Using Tablets and Laptops. In A. Marcus (Ed.), *Design, User Experience, and Usability. Theories, Methods, and Tools for Designing the User Experience, 8517*, 402–412. Switzerland: Springer International Publishing. https://doi.org/10.1007/978-3-319-07668-3_39

Wise, A. F. (2014). Designing pedagogical interventions to support student use of learning analytics. *Proceedings of the Fourth International Conference on Learning Analytics and Knowledge – LAK '14*, Indianapolis, USA, March 24-28, 203–211. https://doi.org/10.1145/2567574.2567588

Appendix 1: Ethics Clearance

UNIVERSITY OF CAPE TOWN
IYUNIVESITHI YASEKAPA • UNIVERSITEIT VAN KAAPSTAD

Faculty of Science
University of Cape Town
Rondebosch
South Africa 7701

E-mail: shari.daya@uct.ac.za
Tel: 021 650-2880

23 May 2019

Mr Kinsley Ndenge
Department Computer Science

RE: Investigating learning analytics for automatically marked programming assessments

Dear Mr Kinsley Ndenge

I am pleased to inform you that the Faculty of Science Research Ethics Committee has approved the above-named application for research ethics clearance, subject to the conditions listed below.

- Implement the measures described in your application to ensure that the process of your research is ethically sound; and
- Uphold ethical principles throughout all stages of the research, responding appropriately to unanticipated issues: please contact me if you need advice on ethical issues that arise.

Your approval code is: **FSREC 47 - 2019**

I wish you success in your research.

Yours sincerely

Dr Shari Daya
Chair: Faculty of Science Research Ethics Committee

Cc: Associate Professor Hussein Suleman (supervisor)

Appendix 2: Informed Consent

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF CAPE TOWN
PRIVATE BAG X3
RONDEBOSCH 7701
SOUTH AFRICA

RESEARCHER: Mr Kinsley Ndenge
TELEPHONE: +27-73-764-4323
E-MAIL: Ndnkin001@myuct.ac.za
URL: http://dl.cs.uct.ac.za/

Informed Voluntary Consent to Participate in Research Study

Project Title: Investigating a learning analytics interface for automatically marked programming assessments

Invitation to participate, and benefits: You are invited to participate in a research study conducted with UCT computer Science Students. The study aim is to investigate learning analytics for automatically marked programming assessments submitted in Vula. I believe that your experience would be a valuable source of information, and hope that by participating you may gain useful knowledge.

Procedures: During this study, you will be interviewed on your experience using the Vula automatic marking tool and might be asked at a later stage to complete a questionnaire in order to evaluate the usability and usefulness of a learning analytics dashboard.

Recording: We may record audio as part of the study. If you object to this, please indicate this below.

Risks: There are no potentially harmful risks related to your participation in this study.

Disclaimer/Withdrawal: Your participation is completely voluntary; you may refuse to participate, and you may withdraw at any time without having to state a reason and without any prejudice or penalty against you. Should you choose to withdraw, the researcher commits not to use any of the information you have provided without your signed consent. Note that the researcher may also withdraw you from the study at any time.

Confidentiality: All information collected in this study will be kept private in that you will not be identified by name or by affiliation to an institution. Confidentiality and anonymity will be maintained as pseudonyms will be used.

What signing this form means.

By signing this consent form, you agree to participate in this research study. The aim, procedures to be used, as well as the potential risks and benefits of your participation have been explained verbally to you in detail, using this form. Refusal to participate in or withdrawal from this study at any time will have no effect on you in any way. You are free to contact me, to ask questions or request further information, at any time during this research.

I agree to participate in this research (tick one box) ☐ Yes ☐ No _________ (Initials)

I agree to the use of properly anonymized audio recordings for research purposes

☐ Yes ☐ No _________ (Initials)

Name of Participant	Signature of Participant	Date
Name of Researcher	Signature of Researcher	Date