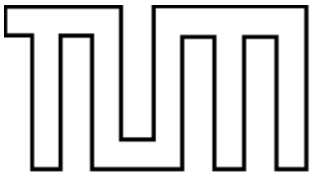
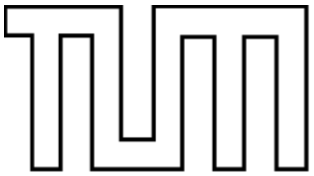


Department of Informatics
Technical University of Munich

Bachelor's Thesis in Information Systems

Development of a student performance monitoring tool based on technical requirements for Information Systems Education

Furkan Gürbüz



FAKULTÄT FÜR INFORMATIK
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Bachelor's Thesis in Information Systems

Entwicklung eines Tools zur Überwachung der Studentenleistung basierend auf technischen Anforderungen für die Wirtschaftsinformatik

Development of a student performance monitoring tool based on technical requirements for Information Systems Education

Author	:	Furkan Gürbüz
Supervisor	:	Prof. Dr. Helmut Krcmar
Advisors	:	Borys Levkovskyi, M.Sc.
		Sophie Heim, M.Sc.
Submission date	:	15.10.2021

Ich versichere, dass ich diese Bachelorarbeit selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

I confirm that this bachelor's thesis is my own work and I have documented all sources and material used.

Garching b. München, den 15.10.2021

_____	_____
Ort, Datum	Unterschrift

Abstract

This Thesis has the goal of defining and implementing a prototype for the learner performance monitoring tool, integrated into the ERP System of the SAP UCC Munich. In the beginning, the requirements framework will be defined. Therefore the requirements are elicited from literature and existing solutions with the same purpose and functionality. For the implementation of the elicited requirements, the ABAP programming language and the SAP UI5 framework are being used. After the implementation, there will be conducted quality assurance and evaluation with the help of a focus group discussion.

Keywords: monitoring tool, e-learning, student performance, requirements engineering, framework, prototyping, ERP-System

Table of Content

List of figures

List of tables

List of Abbreviations

ABAP	Advanced business application programming
ACM	Association for Computing Machinery
AISeL	Association of Information Systems Electronic Library
IEEE	Institute of Electrical and Electronics Engineers Inc
ERP	Enterprise resource planning
GBS	Global Bike Sharing
GBI	Global Bike Inc
REST	Representational state transfer
UI5	User interface for HTML 5
UCC	University Competence Center
CMS	course management system
LMS	learning management system
SUS	System Usability Scale

1 Introduction

1.1 Motivation

The Corona Virus has a major impact on educational institutions like schools and universities, which led to a transformation of a common education to new teaching techniques. Due to the overall situation, e-Learning education gained a lot of importance and most schools had to integrate the e-learning method into their educational methodology (Radha, Mahalakshmi, Kumar, & Saravanakumar, 2020). Even before the virus, universities practiced online platforms like e.g. Moodle, live streams of lectures, and let students access materials without limitations.

E-learning platforms also called learning management systems, or course management systems are defined as "a type of software application that enables instructors to deliver information to students, produce content materials, prepare assignments and tests, engage discussions, and manage distance classes over the Internet" (Mazza & Milani, 2004). By this definition the consideration of a student performance monitoring tool is clear. In order to sustain a high-quality education in an e-learning platform, it is important to consider Bloom's taxonomy of cognitive learning objectives. Bloom's taxonomy defines six categories: Knowledge, Comprehension, Application, Analysis, Synthesis, Evaluation. Hereby the order of the categories also provides information, the complexity is increasing per category (Adams, 2015). Now taken into consideration, that evaluation has a high complexity it shows the need for an additional monitoring tool integrated into CMS, where students and instructors can provide feedback to each other and reflect on their performances.

The SAP UCC provides and improves lectures or training for its customers, with the usage of modern SAP products and technologies. Therefore the team develops and maintains learning material in the matter of different curricula, that teaches the concepts of business processes with the help of SAP-based systems. Hereby the theory is supported with case studies, lecturer´s notes, and slide sets. The SAP UCC uses the data set of a fictive company called GBI so that the learners can identify more with the product and the complexity of the theory is broken down to a more understandable concept. In one of the curriculum, this fictive company experiences a digital transformation process. The company as a traditional manufacturing company, which can not gain any profit from their IoT bikes, changes the whole strategy and the business model to a bike-sharing concept. The learners experience the transformation process and get an insight into the mandatory changes on different modules for every affected aspect of the fictive company – now called GBS. Additionally, the usual ERP system does not fulfill the needs of the new company, that's why it is also important to update it. The curriculum is separated into different learning modules, where the digital transformation process is explained in each affected aspect. The current state of the curriculum does not provide any type of monitoring of the performance of the students. The monitoring tool should provide the learners with information, let the students reflect on their performance in tests/assessments so that a maximum learning effect is satisfied. In addition to that, the instructors should be able to see how well the learners are solving the tasks and give constructive feedback to the learners. The main goal is to gain in the quality of the education overall so that the learners can be prepared for a qualified future workforce.

1.2 Research Question and Methodology

The Bachelor's Thesis is separated into different aspects. On the one hand, the reader gets information about the technical requirements, which are divided into functional, and non-functional requirements, and are conducted by literature, and existing e-learning platforms. This information will be tailored to the needs of the SAP UCC. On the other hand, this thesis will also deal with the actual implementation of the Monitoring Tool, whereupon there will be an evaluation of the finished prototype.

After the implementation of the Monitoring Tool for the existing web-based ERP system, the system will be upgraded with a tool, which helps students and instructors reflect on their performance and improve themselves.

Research Question 1: What are the technical requirements for a Monitoring Tool in an e-learning environment?

First, I will conduct the technical requirements for the monitoring tool, where I gather information from existing literature. Hereby I will be within the context of Webster and Watson (2002 INSERT CITATION). In addition to the literature review, I will also try eliciting requirements out of existing e-learning platforms e.g. Moodle. I will be managing the requirements according to Rupp, Simon, and Hocker (2009) and tailor these to the needs of the SAP UCC. The last step is to create and present a product requirements document according to Teich (2008).

Research Question 2: How can the elicited requirements for a Monitoring Tool in an e-learning environment be implemented?

Here I will extend the existing system with the monitoring tool. For the implementation of the monitoring tool, I will use the gathered requirements of RQ1 and support the infrastructure/concept with a class diagram for better understanding and overview. For the frontend implementation, I will use the SAP UI5 framework, since we are using ABAP as our backend development language. These development languages/frameworks are based on the existing system, which makes integration easier and more consistent.

Research Question 3: How much does the developed Monitoring Tool fulfill the technical requirements?

Lastly, I will evaluate the created prototype to what extent it fulfills the gathered requirements in RQ1. Hereby I will make a survey and/or interview with the SAP UCC Munich team and customers, in order to argue and discuss whether the developed tool meets the requirements for the SAP UCC and if there is still room for improvement.

The Thesis will have a timespan for 5 Months starting at 15.05.21. Firstly I will conduct the literature review from RQ1 with the aim to create the PRD. Additionally, I will familiarize myself with SAP UI5 and ABAP, whereas I have some knowledge about the ABAP programming language from my working student job. After completing RQ1 – probably at the end of June –, I will focus on the implementation of the prototype until early August. Afterward,

I will do the evaluation of the prototype and implement improvements or correct any type of bug fixes. Lastly, I will be writing the thesis itself for about a month.

1.3 Differentiation

The implementation of the prototype will be divided between Henryk Mustroph and me. Hereby it is important to note that Mr. Mustroph is working on his own thesis. His research methodology and field focus on the didactic and the internal logic of the monitoring tool, whereas I am focusing only on the technical requirements of the project.

The implementation process is differentiated within different business process steps between me and Mr. Mustroph. He concentrates on "Materials Management" and "Finance & Controlling", whereas I focus on "Enterprise Asset Management" and "Production Planning".

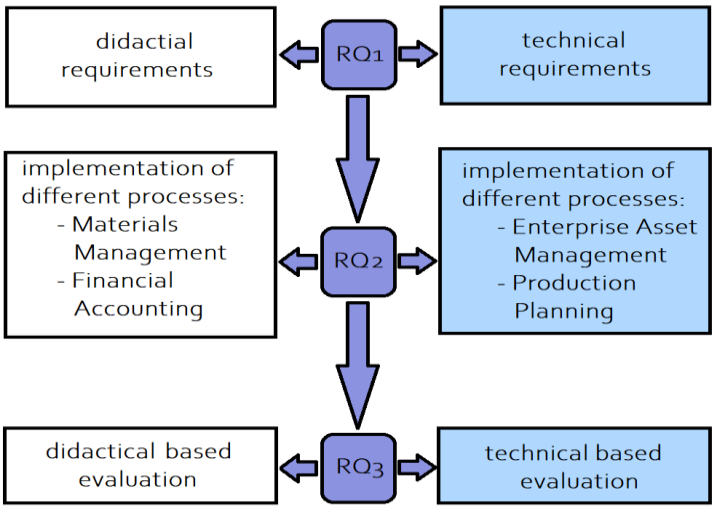


Figure 1: Differentiation
Source: own creation

1.4 Thesis Structure

The thesis has in the following 5 more chapters. In the beginning, I will go into the details of important and relevant keywords, in order to avoid ambiguities inside this thesis. Afterward, I will showcase the requirements framework, which is separated into 2 main areas. The requirements elicitation from literature and existing solutions. Therefore I will go into the detail of the comment section, the Authentication process of the monitoring tool, which directly leads to the Lecturer's or Student's set of actions. Additionally, I will give information about the feedback functionalities. Next, the implementation process will be showcased. Therefore I will not give information about the actual code, moreover, I will talk about the architecture, the database structure, and different process models of the monitoring tool. In the following part, I will go into the details of the evaluation process. This also includes the SAP ACC presentation, where feedback is given from the participants inside this event. The SAP ACC is an event which is sponsored by the SAP and organized by the TUM. Henryk Mustroph and I will present our Monitoring Tool, in order to get feedback and improve our prototype. The feedback will be also added to this thesis. Next to the SAP ACC presentation, I will conduct an evaluation with the SAP UCC employees. Therefore I will create an evaluation model and a survey, which will be adapted from the SUS design and evaluate these results. Lastly, I will be discussing the results of the prototype and additionally giving a future outlook on this prototype. This includes various

rooms for improvement of functionalities and design. The structure of this thesis is showcased graphically in Figure 2.

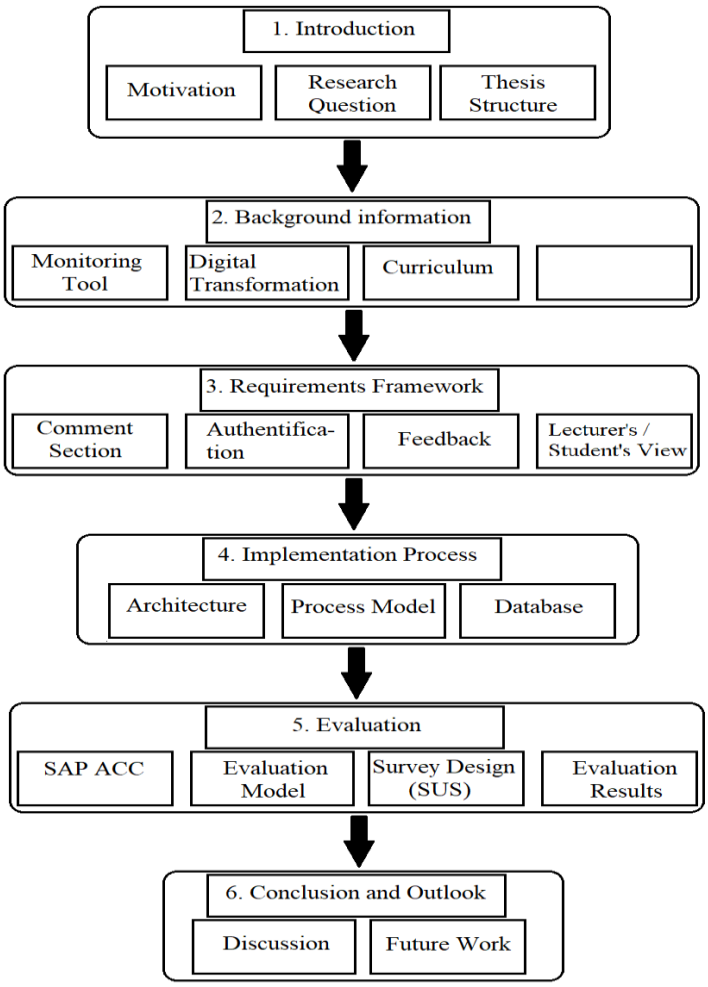


Figure 2: Thesis Structure
Source: own creation

2 Basic Definitions

In the following chapter, I will describe each relevant and important keyword within this thesis. Therefore I will define and give precise information on the keywords. These include the monitoring tool and Curriculum.

2.1 E-Learning System

The E-learning platform is "a type of software application that enables instructors to deliver information to students, produce content materials, prepare assignments and tests, engage discussions, and manage distance classes over the Internet" (Mazza & Milani, 2004). Furthermore, according to Baumgartner, Häfele & Maier-Häfele (2002), there are five main areas in the e-learning platforms, present content, discussion forums, video conferencing and chat, assignments and tests, evaluate and assess student performance (Graf, 2007).

2.2 Monitoring Tool

The purpose of this thesis is the creation of a monitoring tool. Beforehand it is important to gather definitions and relevant aspects of a monitoring tool, in order to gain a specific level of understanding of this topic. A monitoring tool in e-learning educational systems is used to support lecturers in a no face-to-face environment to monitor specific student/course data (Alowayr & Badii, 2014).

The overall purpose is the performance of the analysis. Therefore the data of the current situation of education has to be examined, and additionally, alternatives for its improvement have to be suggested, in order to make an informed decision and carry out the management ensuring the quality of education and science (Lithuania, 2016).

Hereby the important factor is the storage and evaluation of the data. In order to create an effective monitoring tool, the prototype should be able to store all relevant data that the student or the lecturer wants to evaluate. On the one hand, the relevant data for the lecturer would be the actual input of the student, the reached scores from the curricula, and the learning progress. On the other hand, the relevant data for the student would be the correct answers, his right/wrong input, and the assessment of the lecturer.

2.3 Curriculum

In order to maintain the educational quality, it is important to support the theoretical concepts with different curricula. These curricula have a major impact on the student's learning effectiveness. A Curriculum "is often defined as the courses offered by a school […]". More specifically a "curriculum typically refers to the knowledge and skills students are expected to learn, which includes the learning standards or learning objectives they are expected to meet; the units and lessons that teachers teach; the assignments and projects given to students; the books, materials, videos, presentations, and readings used in a course; and the tests, assessments, and other methods used to evaluate student learning […]" (Glossary, 2015).

3 Requirements Framework for a monitoring tool

In this chapter, I build up to two different main methodologies – requirements elicitation from literature and requirements elicitation by existing solutions – the requirements document for the prototype. Hereby it is important to mention the potential problems in agile project development. According to the international survey conducted by Stefan Wagner, most projects suffer because of unclear/unmeasurable non-functional requirements and underspecified requirements (Wagner, Fernández, Felderer, & Kalinowski, 2017). Those can increase the overall development time, in fact even dump the whole project in worst-case scenarios. This survey undermines the importance of a clear requirements document, in order to create a prototype.

3.1 Methodology

As mentioned in the previous section, I build up two different main methodologies. For the requirements elicitation from literature, I will use the literature review framework from Brocke. Thereby Brocke structures the literature review into 5 phases as follows: The first step is the definition of a review scope. Next, the topic should be conceptualized, which includes the definition of key terms – as in chapter 2 . Afterwards, a literature search needs to be conducted. In the fourth step, the gathered literature is analyzed and synthesized. Lastly, the research agenda is created by the synthesized literature (Brocke et al., 2009; Müller-Bloch & Kranz, 2015). The overall structure of these phases can also be seen in the Colum "Actions" of Table 1.

The results of the literature review according to Brocke et al. are gathered in Table 1. In the first phase, a review scope has to be defined. Therefore, in order to maximize the quality of the research, only databases that are most suited have been searched, which are the IEEE Xplore Digital Library, ACM, AISeL, and Scopus. The selection of these databases is based on their reputation and recommendations. The second step was the topic conceptualization, for which I defined specific search queries, that suit the topic of this thesis.

The search queries:

- ("E-Learning" OR "Learning Management System" OR "Course Management System" IN (abstract OR title)) AND ("Monitoring Tool" IN (abstract OR title))

- ("E-Learning" OR "Learning Management System" OR "Course Management System" IN (abstract OR title)) AND ("monitoring tool" IN (abstract OR title)) AND (("requirements" OR "technical requirements") IN abstract)

In order to cover all options, I created in the first and the second query an OR operation with e-learning, learning management system, and course management system, because all of these three descriptions are synonyms, which are commonly used for e-learning environments. I decided to search these descriptions inside the abstract or the title, because of their importance. Then I combined the OR operation with an AND operation with the monitoring tool, which is also one of the important keywords in this thesis. Lastly, I added a third AND operation in the second query, where I searched for technical requirements inside the abstract.

The queries and the literature research helped me to create a research agenda – formulation of the requirements – with a total number of 24 requirements within 6 different categories, which will be explained and showcased in the next chapter.

Actions	Results
Review Scope Define the databases to be queried	IEEE Xplore Digital Library, ACM Digital Library, AISeL, Scopus
Topic Conceptualization Only consider journal and conference articles using 2 queries.	479 total papers have been found
Literature Review Scanning of titles and abstracts	35 papers are considered relevant
Literature Analysis and Synthesis Reading the remaining papers	10 papers contain statements that can be transformed into requirements
Research Agenda Formulation requirements	24 requirements of 6 categories

Table 1: Literature Review Methodology
Source: own creation

3.2 Case Studies in Global Bike Sharing

In this chapter, the overall process of the case studies and the idea of GBS will be explained. As stated in the introduction part, the SAP UCC tries to introduce and teach its customers the overall process of digital transformation, in combination with SAP technologies. Therefore they created materials, which consists of case studies. These case studies are curricula, where the theory is explained and visualized with practical examples, that's why the SAP UCC created a fictive company named Global Bike Inc. GBI was a company with a traditional business model, which was producing and selling normal bikes. Nevertheless, this fictive company goes through the digital transformation process, because the company could not generate a valuable profit from this business model. This transformation process affects, creates, and replaces business processes from the traditional business processes, as illustrated in Figure 3.

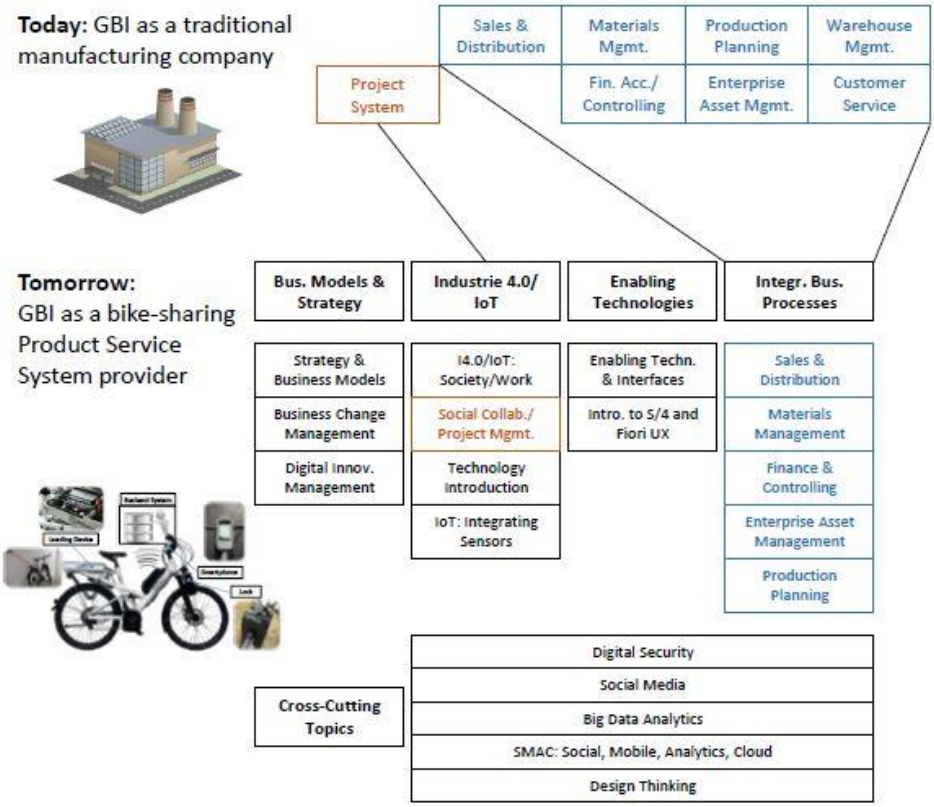


Figure 3: Modules of Global Bike Company
Source: (UCC)

In order to understand the overall process, it is important to go through the case studies. These are structured in different tasks, or concepts, which also have steps that the student/learner conducts. These steps require user inputs, which can be letterals, checkboxes, or selections, in order to fulfill the case study. Nevertheless, the learner can make wrong inputs, which has the consequence that the upcoming steps can not be fulfilled, because of the particular mistake. That's why the lecturer always has to control and monitor the student, in order to avoid incomplete case studies conducted by the student. After the learner conducted the case study, on the one hand, the Lecturer assesses and grades the Student, and on the other hand, the student can also give feedback to the lecturer about the overall experience on the conducted case study. The structure and relationships inside the process of the SAP UCC teaching mechanism are visualized in Figure 4.

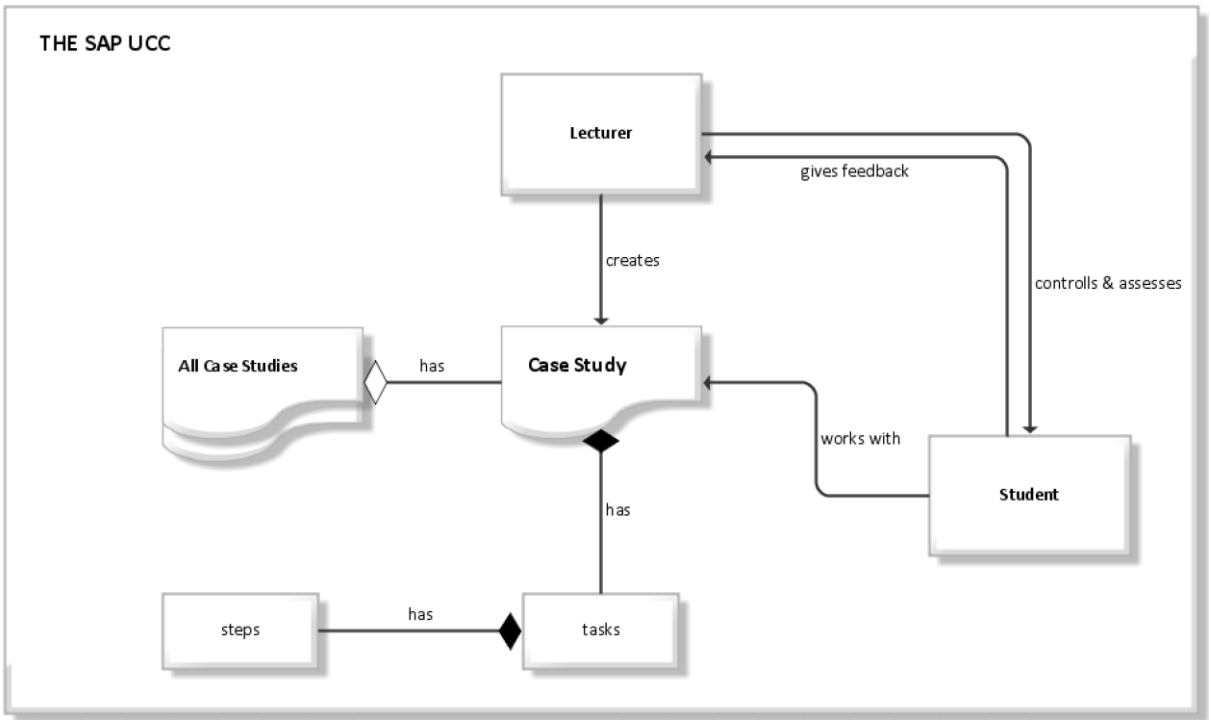


Figure 4: SAP UCC Case Studies
Source: own analysis

3.3 Eliciting Requirements from Literature

In this chapter, the requirements gathered from the literature review will be analyzed and synthesized. Therefore different categories are created, in order to sustain readability and improve the overall structure of the requirements conceptualization.

3.3.1 Organizational Requirements

The organizational requirements are one of the most important categories because this category provides the prototype for the differentiation between different use cases.

In traditional learning environments there coexist 2 parties, which are on the one hand the Lecturer and on the other hand the student/learner. This directly leads to the consequence that the prototype should also have the functionality of 2 different use cases to maintain the core principle of the traditional learning environment. Another reason for the separation of the different views is in order to sustain a quality learning experience, because of the added value of functionalities as described in the following chapter 3.3.2 “Requirements for the Set of Action Default User” and 3.3.3. “Requirements for the Set of Action Admin User”. The requirement for the access control is that *“the system should differentiate between Admin and Default user, who has different levels of restrictions and adapt the UI accordingly”.*

The use case UC_OR1 for the OR01 requirement is created in table 2, to get a better understanding of the concept and functionality. Therefore it is important to note that, the actors are divided into Lecturer and Student, whereas some steps have identical actions regardless of the logged-in user. After the user is doing the trivial steps from 1 and 2. The system checks whether the input values given by the user are correct. If this is not the case the system gives out an error message with the request to try the login process again. If the input values are

correct then the system redirects the user to the overall landing page, where another if statement is defined, which again checks for the logged-in user, and redirects the user based on the login details. Therefore it is given that the SAP UCC uses a predefined identification number for students, which starts with the letterals “GBS” and a 3 digit number attached with a “-”. These predefined identification numbers help the implementation of the requirements because it is easy to check within an if statement.

Use Case	Login into the system (UC_OR1)		
Goal	Users can log in to the system and the system can detect the currently logged-in user.		
Preconditions	The User knows his/her login details		
Actor(s)	Lecturer/Student		
Main Success Scenario	1.	Lecturer/Student	types in the username
	2.	Lecturer/Student	types in the password
	3.	System	checks input values with the data values saved in the database: **If** input values are **true** redirects to the landing page; **Else** give out an error message
	4. Case: **false**	Lecturer/Student	starts from step 1 again.
	4. Case: **true**	System	checks the login details: **If** the username starts with GBS, then redirect to the student landing page; **Else** redirect to the lecturer landing page

Table 2: Use Case organizational 1
Source: own analysis

3.3.2 Requirements for the Set of Action Default User

It has come clear that the e-learning environment differs significantly from traditional learning variants, whereas the underlying principles of the assessment of student performance are not very different. That's why the Pennsylvania State University could define principles, in order to create a guideline for e-learning assessment, which is visualized in Figure 5 (Meyen, Aust, Bui, & Isaacson, 2002).

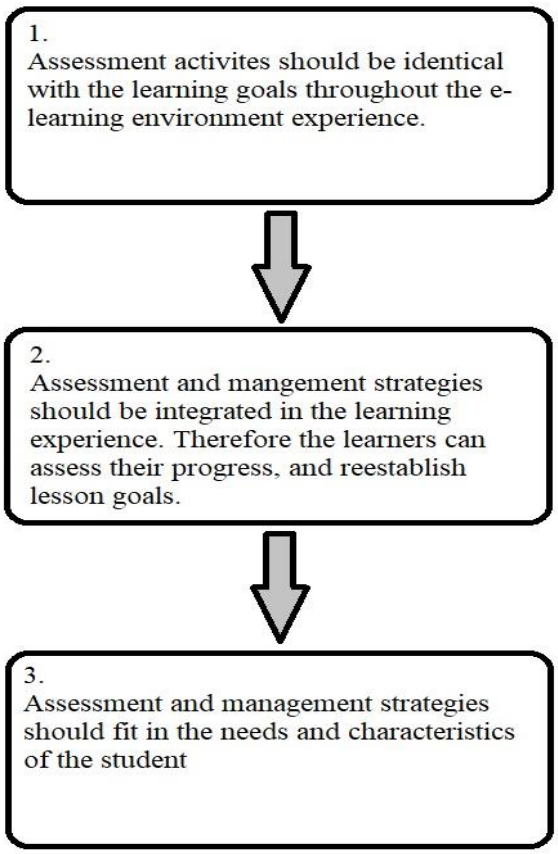


Figure 5: Guideline for e-learning assessment
Source: own analysis

Based on the guideline, the first point is that the learning goals in e-learning should be identical to the assessment activities from traditional learning environments. This is actually covered by the SAP UCC Curriculums, that’s why this requirement is set as a pre-condition for the prototype.

The second guideline is the assessment and management strategies that should be integrated into the learning experience, with the goal that students can assess their progress and reestablish their lesson goals. In other words, it is important that the student can assess themselves, in order to increase the quality of the learning objectives, because he/she is recapturing the whole case study, which directly leads to a more in-depth view into the assessment. Hereby it is important to mention, that the overall assessment with the shown data should not be on the home screen. This is mandatory for readability and structural reasons. Because of that, the integration of a button, which leads to the visualization of the data should be integrated into the prototype, and

the belonging requirement is that *"The default user should be able to confirm and graphically see his/her results by pressing a button."*

The last guideline is the assessment and management strategies that should fit the needs and characteristics of the distance learner. Because of simplicity reasons, the needs of the learner can be interpreted as the selection of the Case Studies. This has the consequence that the user can select and personalize his/her mandatory needs for the assessment. Therefore he/she can select a case study, whereupon the user can self-assess themselves. The belonging requirement is that *"The default user should be able to select a specific case study from a drop-down menu"*. Additionally, it is only logical to integrate the functionality that the learner can deselect his/her selection. *"The default user should be able to delete (UI based) the selection of SoAD01."*

The use case UC_D1 for SoAD01 is created in table 3, to get a better understanding of the concept and functionality of the important requirements for the default user. The overall aim of the use case is, that the student wants to select the desired case study, in which he/she wants to monitor himself/herself. The precondition hereby is the use case of the organizational requirements (UC_OR1), which describes the login and redirection process of the user. The overall steps are easy to follow. From steps 1 to 3 the student basically selects the case study from dropbox, whereas in step 4 is the trigger point for the if statement in step 5, where the selection of the user is checked. If the selection is invalid the system gives out an error message, else the selection was valid and the user gets redirected to the next page, where the student's data is showcased.

Use Case	The student selects a case study (UC_D1)		
Goal	The student wants to select the desired case study		
Preconditions	The student opened the website and logged himself/herself in (UC_OR1).		
Actor(s)	Student		
Main Success Scenario	1.	Student	clicks on dropbox
	2.	System	showcases hardcoded case study values
	3.	Student	clicks on desired case study
	4.	Student	clicks on arrow
	5.	System	checks entry values: **If <u>true</u>** redirect to the next page; **Else** give out an error message

Table 3: Use Case Default 1
Source: own analysis

After the use case described in the previous sequence (UC_D1), the use case UC_D2 for SoAD02 is showcased in table 4. This use case describes less functionality, but still is one important factor of the monitoring tool. The self-assessment of the student is described in the use case. The student has done the actions of UC_d1 and gets redirected to this page. Here the system loads the data from the backend and showcases it for the user, in form of diagrams, statistics, and right or wrong assessments, where the student can scroll through the data.

Use Case	The student sees the data (UC_D2)		
Goal	The Student wants to see the data of the selected case study (which he/she selected in UC_D1)		
Preconditions	The Student has done the actions of UC_D1, and the if case was identified as true by the system!		
Actor(s)	Student		
Main Success Scenario	1.	System	loads data from backend from specific student and case study
	2	Student	scrolls through the data and assesses his results

Table 4: Use Case Default 2
Source: own analysis

3.3.3 Requirements for the Set of Action Admin User

As stated in the previous section – "Organizational Requirements" –, the prototype differs between two users. The first user is the default user, where the restrictions and set of actions were defined in chapter 3.3.2. The second user is the admin user, who has more rights, other sets of actions, and different UI. The logic behind the monitoring tool hereby is not to completely replace the lecturer, instead, the lecturer still is an important role inside the monitoring process. Nevertheless, the prototype helps the lecturer to assess and control the student's activity, results, and feedback.

According to Meyen (2002), there are nine principles of assessment, which have to be taken into consideration for a web-based assessment tool. One of them would be the consideration of whom the assessment is conducted. Hereby the Lecturer should be able to assess/check the results of the student, in order to monitor the performance of the student and get a report from the monitoring tool, which increases the overall quality of education. Therefore the lecturer should be able to select the student, which he wants to monitor. For that, he needs to have the opportunity to select a specific student, *"the admin should be able to fetch students data from the database with a dialog (calling students id)"*. After the selection of the student, the lecturer should also be able to select the case study. Wherefore another requirement arises *"The admin should be able to fetch data of selected case study from a drop-down menu"*. Accordingly to the selection of the entries, the Lecturer should also be able to delete his entries from the dropbox and dialog *"Admin can delete the entries (UI based) from SoAA01 and SoAA02"*. The AAST required from the monitoring tool, to be able to detect the student's activity, which should be saved in the database, and be able to fetch by Users, that have permission to monitor the activity (Daif & Rizkaa, 2013).*"The admin can monitor the activity of the student. Therefore every SoAD needs to have a unique identifier, which needs to be saved in the database"*.

In order to understand the overall concept and functionality of the important requirements for the lecturer, the use case UC_A1 for SoAA01 & SoAA02 is created in table 5. Analogous to the student, the lecturer should also have done the login process correctly in UC_OR1. Additionally, the lecturer gets redirected to this page, which is slightly different than the students' page. The Lecturer has more rights, which means he/she can monitor all default users, whereas the student could only monitor himself/herself within different case studies. In the first step, the lecturer can click on the upper dropbox and the system showcases all the students, loaded from the backend. Afterwards, the lecturer can click on the student, who he/she wants

to monitor. Next, the lecturer clicks on the lower dropbox and selects the case study, in which he/she wants to monitor the selected student. The last step, which needs to be done by the user is to click on the button, which triggers the if statement from step 7, where the entry values are checked.

Use Case	Lecturer Selection of Student & Case Study (UC_A1)		
Goal	The Lecturer wants to select a specific student within a specific case study		
Preconditions	The Lecturer opened the website and logged himself/herself in (UC_OR1).		
Actor(s)	Lecturer		
Main Success Scenario	1.	Lecturer	clicks on the first dropbox
	2.	System	loads data from the backend
	3.	Lecturer	clicks on desired student ID
	4.	Lecturer	clicks on the second dropbox
	5.	System	showcases hardcoded case study values
	6.	Lecturer	clicks on desired case study
	7.	Lecturer	clicks on the arrow button to go to the next page
	7.	System	checks entry values: **If <u>true</u>** redirect to the next page; **Else** give out an error message

Table 5: Use Case Admin 1
Source: own analysis

Additionally, the use case UC_A2 for SoAA03 has been generated in table 6, which is basically analog to the UC_D2.

Use Case	Lecturer visualized the data (UC_A2)		
Goal	The Lecturer wants to see the data of the selected student and case study (which he selected in UC_A1)		
Preconditions	The Lecturer has done the actions of UC_A1, and the if case was identified as true by the system!		
Actor(s)	Lecturer		
Main Success Scenario	1.	System	loads data from backend from specific student and case study
	2	Lecturer	scrolls through the data and monitors the results

Table 6: Use Case Admin 2
Source: own analysis

3.3.4 Requirements for the Comment Section

Meyen (2002) stated that it is important to transform from the perspective of periodic exams and graded activities, where the overall discussion of the activity will be or will not be discussed in the course, to consider it in the context of e-learning where students can view their relationship with the instructor in asynchronous or synchronous communication. This will lead the assessment of graded activities to moreover a continuous process (2002). This shows why the communication between student and instructor becomes important in e-learning environments and can not be overseen. Meyen et. Al has citated Morgan and O'Reilly, who describe the assessment options within five different categories (also see Figure 11). "Ungraded activities and feedback built into study materials; Self-assessment quizzes and tests that allow students to check their own learning; Formal feedback on assignments from instructors, peers or work place colleagues or mentors; Informal dialogue with instructors, peers or others; Ungraded tests that prepare learners for formal grade assessments" (2002). The comment section hereby falls into the category of the informal dialogue, because the comment sections are supposed to interact in a non-formal – but still intellectually stimulating – with the lecturer or vice versa. That's why the prototype has the requirement for the benefit of the lecturer that "The admin should be able to comment inside a comment section on the student's solution (asynchronous interaction)". Additionally, the student should be able to write a new comment to the lecturer or respond to the comment of the lecturer *"The default user should be able to respond to the comment of the admin or write a new comment to the admin (asynchronous interaction)"*. An important factor hereby is the visibility of the interaction between student and lecturer, which should be discrete and only for the concerned parties, wherefore the requirement "*The Comments inside the comment section are only visible for concerned users"* is created.

The Use case UC_CA1 is generated for better understanding. Here the lecturer can write a comment of different patterns of the specific student in a specific case study. These are the task performance, the activity behavior, and the learning progress. After the lecturer has decided on what to comment for the student, he clicks on the specific comment field and types the comment. After clicking on the button the comment is saved into the database with the specific receiver id and on the pattern id.

Use Case	Comment functionality for the lecturer (UC_CA1)		
Goal	The Lecturer wants to write a comment for the student within a case study, which were selected in UC_A1		
Preconditions	The Lecturer has already done UC_A1		
Actor(s)	Lecturer		
Main Success Scenario	1.	Lecturer	can write a comment on the student's task performance, or activity behavior, or learning progress for the selected case study and student
	2.	Lecturer	clicks on the comment section field and types in the comment
	3.	Lecturer	clicks on the send button

	4.	System	saves the typed-in comment into the database, with the specific user id and case study id.
	5.	System	**If** an error occurred during step 4, then the error message box is visualized; **Else** success message box is visualized

Table 7: Use Case Admin 3
Source: own analysis

The next use case is analogous to UC_C1, except that the actor is the student (instead of the lecturer) which has the consequence that the student does not have the opportunity to decide on what specific topic the comment falls in. The comment is more directed to the overall case study, without any specialization.

Use Case	Comment functionality for the student (UC_CD1)		
Goal	The Student wants to write a comment for the lecturer within a case study, which were selected in UC_D1		
Preconditions	The Student has already done UC_D1		
Actor(s)	Student		
Main Success Scenario	1.	Student	clicks on the comment section field and types in the comment
	2.	Student	Clicks on the send button.
	3.	System	saves the typed-in comment into the database, with the specific case study id.
	4.	System	**If** an error occurred during step 3, then the error message box is visualized; **Else** success message box is visualized

Table 8: Use Case Admin 4
Source: own analysis

3.3.5 Requirements for the Feedback

Feedback is defined as "[…] providing a person information on how he or she performed in light of what he or she attempted." (Meyen et al., 2002). Additionally, as seen in the previous sequence (3.3.4. Requirements for the comment Section) Morgan and O'Reilly created 5 different categories (also see Figure 6), whereas the feedback falls into the category of "Formal feedback on assignments from instructors", which also adds value to the continuousness of the assessment process. Because of these reasons the prototype has a bidirectional feedback opportunity which consists between lecturer and student. Therefore the requirements *"the*

admin should be able to give a total grade/points for the selected default user within the whole case study. This feedback should be only visible to the concerned user. Therefore the points/grade has to be saved with the default users id" and *"The default user should have once permission to give feedback to the admin for a specific case study. This feedback should be sent (and saved in the database) to the admin's feedback page"* are created.

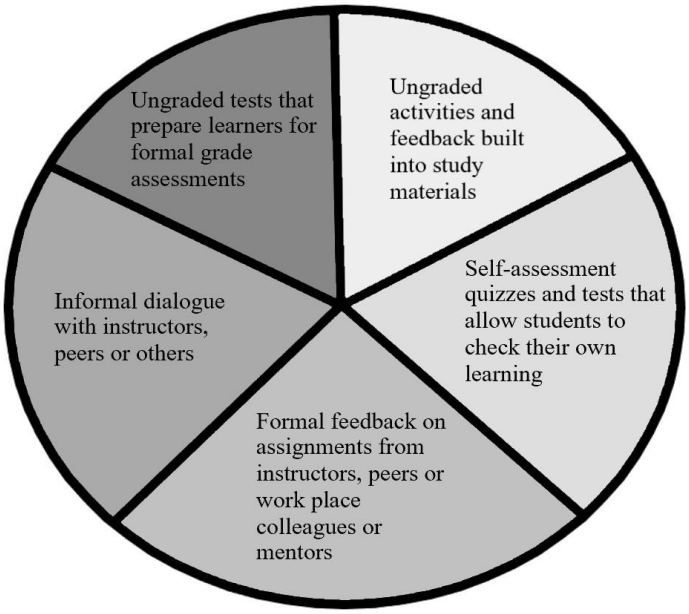


Figure 6: Assessment categories
Source: own analysis

3.3.6 Non-functional requirements

Usability has a major impact on the creation of a user-friendly tool. Therefore the developer should take the human-computer interaction (HCI) or usability engineering perspective, in order to generate the usability requirements (Röder, 2012). According to Röder, "Software engineers often do not take usability features into account systematically, and, in particular, do not consider them "first-class citizens" during requirements-related activities" (2012). Therefore Röder defines a usability pattern, in order to systematically generate a user-friendly tool – see Figure 7. The Undo pattern allows users to revert and delete specific actions. Furthermore, the auto-save, where changes are saved automatically without any user action. Next is the run in background pattern, which "allows users to have the system perform long-running tasks in the 'background', and to meanwhile continue working with the system". The Live validation validates user input while the user is typing and give direct feedback. The progress display informs users about the progress of long-loading tasks and algorithms. Next is the recycle bin, which allows users to move data to a virtual recycle bin instead of deleting the data directly and permanently, in order to restore the data if needed (Röder, 2012).

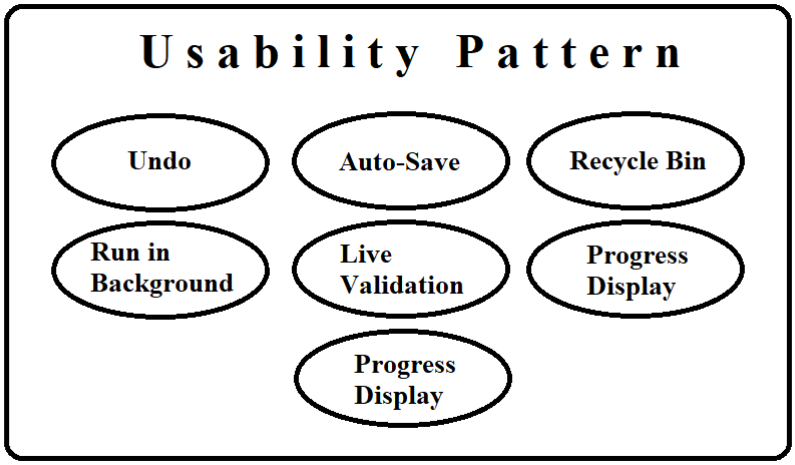


Figure 7: Usability Pattern
Source: own analysis

3.3.7 Preliminary Results

Every discussed requirement in the previous sequences is visualized and gathered in Table 9. Therefore the requirements are grouped into their different categorizations and assigned a unique requirement identifier, in order to sustain readability and simplicity. The requirement identifier is created within one unique key value – whereas the letterals stand for the associated categorization, and the numerals for the counter, which starts from 01 –, and a short title.

Requirement Identifier	Requirement Description
/OR01/ Access control	The system should differentiate between Admin and Default user, who has different levels of restrictions and adapt the UI accordingly
/SoAA01/ Fetching data of individual default user	The admin should be able to fetch students data from the database with a dialog (calling students id)
/SoAA02/ Fetching data of case study	The admin should be able to fetch data of selected case study from a drop-down menu
/SoAA03/ Visibility of data	The admin should be able to confirm and graphically see the data by pressing a button
/SoAA04/ Deletion of Selection	Admin can delete the entries (UI based) from SoAA01 and SoAA02
/SoAA05/ Activity Control	The admin can monitor the activity of the student. Therefore every SoAD needs to have a unique identifier, which needs to be saved in the database.
/CS01/ Admin Comment	The admin should be able to comment inside a comment section on the student's solution (asynchronous interaction).

/CS02/ Default User Comment	The default user should be able to respond to the comment of the admin or write a new comment to the admin (asynchronous interaction).
/CS03/ Visibility	The Comments inside the comment section are only visible for concerned users
/SoAD01/ Selecting Case Studies	The default user should be able to select a specific case study from a drop-down menu.
/SoAD02/ Visibility of Selection	The default user should be able to confirm and graphically see his/her results by pressing a button.
/SoAD03/ Delete selection	The default user should be able to delete (UI-based) the selection of SoAD01.
/SoAD04/ Overall Success rate	The default user should be able to see the overall success/failure rate by clicking on a button.
/FB01/ Admin feedback	the admin should be able to give a total grade/points for the selected default user within the whole case study. This feedback should be only visible to the concerned user. Therefore the points/grade has to be saved with the default users id
/FB02/ Default User feedback	The default user should have once permission to give feedback to the admin for a specific case study. This feedback should be sent (and saved in the database) to the admin's feedback page
/FB03/ Visibility feedback page (Admin)	The admin should be able to see all the feedback gathered on the feedback page (send by FB02).
/FB04/ Visibility feedback page (Default user)	The default user just should see one feedback for each case study on his/her feedback page (from FB01).
/NF07/ Usability	The user should be able to navigate throughout the tool within 3 steps

Table 9: Requirements from Literature Review
Source: own creation

3.4 Eliciting Requirements from Existing Solutions

In this chapter, the requirements gathered from existing solutions like Moodle, Artemis, and Campus Tum will be analyzed. It is important to note that the requirements from the literature review can also occur inside some existing solutions, which will be shortly mentioned if this case happens. Additionally, requirements like accessibility, an event of failure, response time,

and language are seen in all of the mentioned solutions. The only one created category hereby is the non-functional requirements.

3.4.1 The Sap UCC Global Bike Application

The overall tool should be based on the SAP Fiori design guidelines, which are role-based, adaptive, coherent, simple, and delightful (SAP). These guidelines affect the experience as a developer and a user, which basically means that the guidelines reflect from the developer to the user *"The tool should maintain the overall design concept of SAP FIORI"*. The overall UI concept is also visualized in figure 8 based on a sample SAP Fiori application, and marked with general UI SAP Fiori aspects. This overall UI design is also implemented in the application of the SAP UCC, which also contributes to this UI design factor.

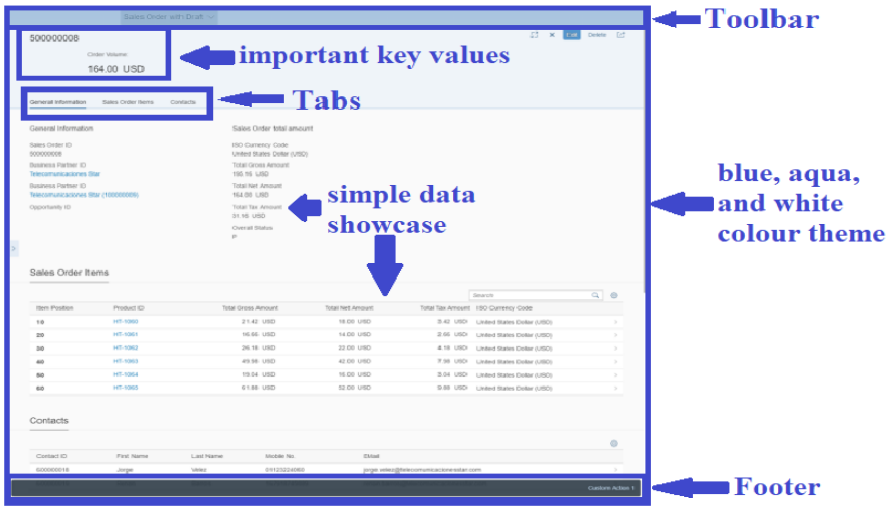


Figure 8: Basic Fiori UI Sample
Source: adapted from quanto-solutions (quanto-solutions, 2021)

The adaptiveness of the tool means that it can adapt itself to different use cases and devices. One use case would be the accessibility for the tool *"The tool should be accessible with web browsers (Firefox, Chrome, etc.)"*. This also increases the simplicity of the tool, because of simple access without downloading or updating any extern program – except for an actual web browser, which is normally given on every device. As in the UI design logic, the adaptiveness is also covered from the SAP UCC application, which is accessible via an actual internet browser.

Additionally, in order to generate a user-friendly tool, it is also important to define the click-flow of the User. Therefore different use cases in the previous chapters have been generated, in order to illustrate the click-flow within the use case.

3.4.2 Moodle

Moodle is a well-known LMS within students and universities. It can be used to give access to lecture material, evaluations, and also quizzes. Hereby Moodle also tries to implement the functionality of interaction between the users, with specific forums and discussions. Nevertheless, this functionality does not support, a one-to-one conversation between students

or student-lecturer, instead it only has a forum with all participants. So the overall idea of having an interaction is also seen in Moodle, as in the monitoring prototype. Furthermore, it is important to note that Moodle also differentiates between the student, and lecturer, which underlines the importance of the OR01 requirement. The average response time of Moodle has been tested and set as less than 5 seconds, and the overall availability of the tool is 99% of the time.

3.4.3 Artemis

Artemis is a tool, which is also popular among students at the TUM. It is mostly used for exams, graded and non-graded homework, or activities. The Tool of course separates between student and lecturer login and can identify what current user is logged in, what supports the importance of the OR01 requirement. Additionally, it has an assessment functionality, which underlines the requirements for the lecturer SoAA01 & SoAA02, and also for the student SoAD01. The average response time of Artemis has been identified and set as less than 5 seconds, and additionally, the overall availability of the tool is 99%.

3.4.4 Preliminary Results

The additional requirements - gathered from existing solutions - which are discussed in the previous sequence are visualized and gathered in the following Table 10. Therefore the requirements are grouped and identified as in Table 9.

Requirement Identifier	Requirement Description
/NF01/ Accessibility	The tool should be accessible with web browsers (Firefox, Chrome, etc.)
/NF02/ Response Time	The average response time should be less than 5 seconds
/NF03/ Availability	The tool should be available 99% times
/NF04/ Language	The tool should be in German and English
/NF05/ Event of Failure	When the tool crashes, it should be able to back up to its previous state
/NF06/ Design	The tool should maintain the overall design concept of SAP FIORI

Table 10: Requirements from Existing Solutions
Source: own creation

3.5 Results Evaluation

In the previous chapters, requirements elicitation and synthesis have been conducted. Therefore a total of six categories and 24 requirements have been created, which are the non-functional requirements, requirements for the comment section, requirements for the feedback, set of actions default user, set of actions admin user, and lastly organizational requirements. In the following Figure, all requirements and categories are visualized.

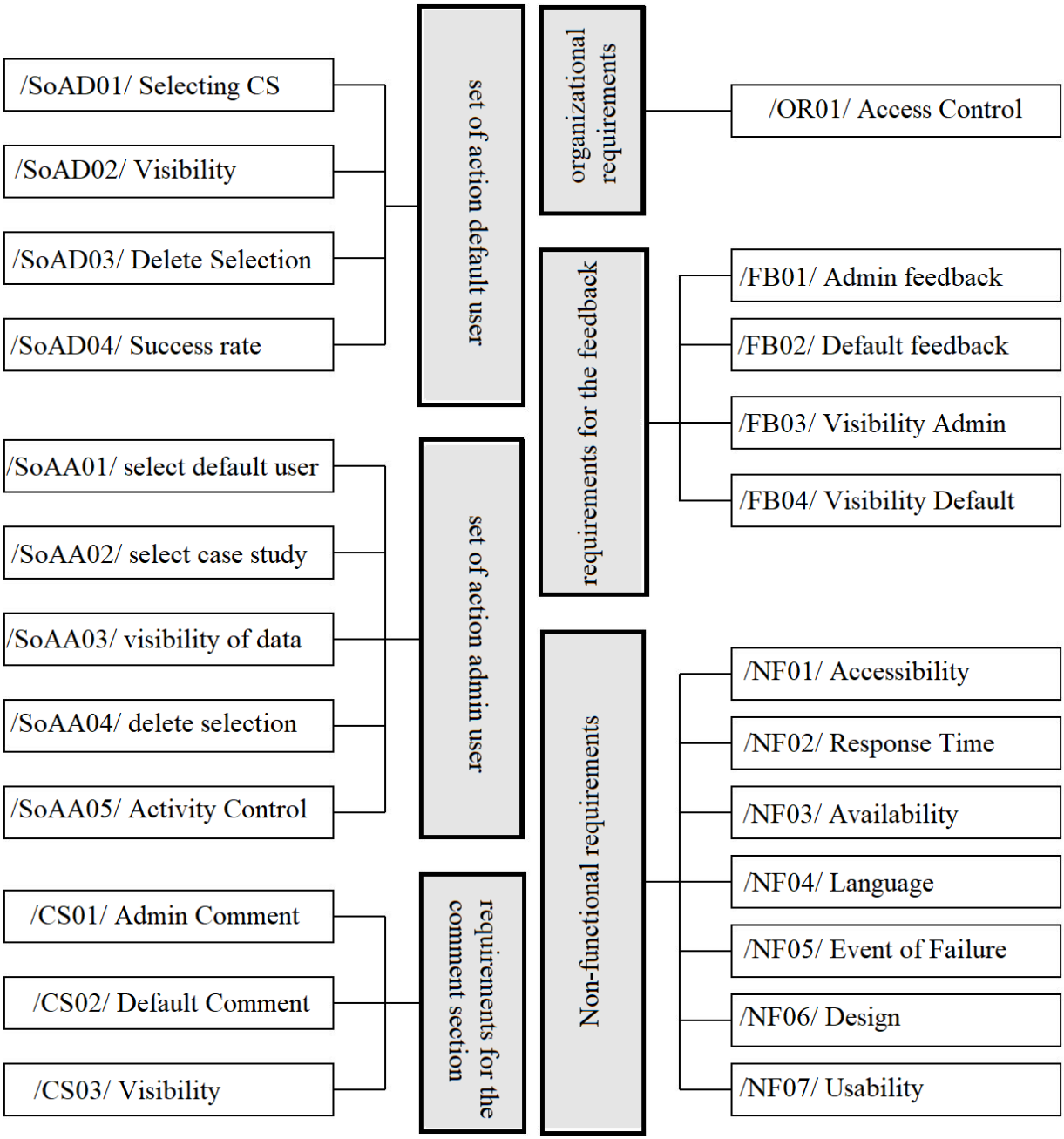


Figure 9: Requirement Results
Source: own analysis

4 Implementation of a Student Performance Monitoring Tool

In this chapter, the elicited requirements from chapter 3, will be described and implemented into the prototype. Therefore the overall structure of the following chapters is identical to the previous sequences. More specifically, the development process will be explained in detail in order, to comprehend the process.

4.1 Methodology

There are many methods for the development process of software, Waterfall, Extreme Programming, and SCRUM. Nevertheless, an important factor in modern software development is to be agile. This allows being able to deliver fast, acknowledge change requests, change quickly, and change often (Cohen, Lindvall, & Costa, 2003). Additionally, Beck et al. defined a total of 12 principles for agile software development, which also supports some values from Cohen. The change requests are defined at Beck et al. as "Welcome changing requirements, even late in development. Agile processes harness change for customer's competitive advantage" (Beck et al., 2001). Next to the principles, it is important to accept the term, which is formulated by Schwaber "[…] the development process is unpredictable" (2003). The key principles of Scrum are small working teams that maximize communication and minimize overhead. Furthermore, the adaptability to technical or market changes to ensure the best possible product. The construction of executables, that can be "inspected, adjusted, tested, documented, and build on" (Cohen et al., 2003). Additionally, the teamwork, and portioning of the work, into low coupled packets. The constant testing and documentation of the product, while the implementation process. Lastly, the ability to declare a product to done when required (Cohen et al., 2003).

As Scrum is more of a widely used method, the implementation process of the monitoring tool will also be done with the Scrum methodology. In order to get a better understanding of the scrum method, the Scrum lifecycle is visualized in Figure 10 and will explain more in detail in the following sequences.

The scrum lifecycle model starts with the pre-sprint planning, where the functionalities and features are depicted from the product backlog, with the important factor that the depicted features are collected inside the sprint backlog, which is a set of prioritized functionalities, that need to be implemented in the following sprint. The Product Backlog, where the features are depicted, is basically a whole set of functionalities that are requested from the software (Cohen et al., 2003).

The next step inside the process is the Sprint, which has a time span of 2 to 4 weeks – time span can vary. The Sprint realizes the implementation of the features gathered in the sprint backlog. It is important to note that the sprint backlog does not change during this process. It is not possible to add functionalities to the sprint backlog. The developers can select the features from the sprint backlog, that they want to implement. Additionally, the Sprint has a Daily Scrum Meeting, which is essential for the overall success of the Sprint. The daily meetings enhance the overall communication of the team members and customers, and additionally informs all participants, who are involved in the project, about the status of the project, identify problems encountered, and keep the participants active and motivated on the common goal (Cohen et al., 2003).

Lastly, the Post-Sprint meeting, where after the sprint a meeting is held to analyze the project progress and visualize the current implemented software (Cohen et al., 2003).

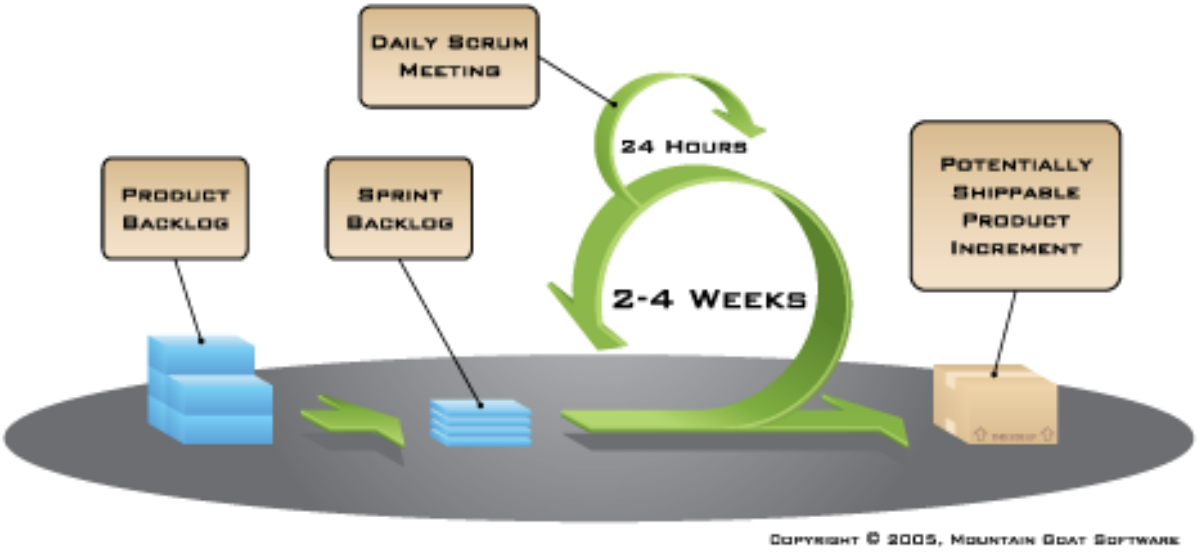


Figure 10: Typical Scrum Lifecycle model
Source: adapted from mountain goat software (Software, 2021)

Next to the Scrum method, is the Kanban methodology, which is also an important factor within the development process. The Kanban has four principles, which are the start with what must be done now; changes are made gradually; respect current roles and responsibilities; encourage leadership at any hierarchical level (Mircea, 2019). Next to the principles, the best practices for implementing the Kanban Method into a team is the visualization of the work, where the team can see all steps of the project in different categories, or columns. Therefore the Kanban Board should be used, and is structured as follows. The board has the columns Future, Next, Requirements Analyzation, Design, Development, Testing, Deploying, and Done. Inside the columns, except in the Future column, all elements should be ordered based on values that are set by the team, which can be importance or difficulty (Mircea, 2019). Another important practice is to make process policies explicit, so every team member has insight to all the processes, policies, and practices, which helps the overall communication and active collaboration of the team members (Mircea, 2019).

Because the overall implementation of the prototype will be done with Mr. Mustroph, the overall conclusion was to use a hybrid method, which has the sprint cycles of the SCRUM methodology combined with the Kanban Board of the Kanban methodology (Bass, Pejcinovic, & Grant, 2016). This hybrid method helps the overall management progress, and helps us to define explicitly tasks and responsibilities for each team member. According to Bass et al., if the Kanban Board is combined with additional meetings with the team, it can lead to positive pressure, which ensures the progress of the project (2016).

For the visual representation of the Kanban Board we used the Atlassian software Trello. The other optional software were Jira, and Wrike. A traditional panel on the wall with sticky notes was not taken into consideration, because of the overall corona restrictions and lack of flexibility. Trello is a web-based project management application, which provides means for creating online Kanban boards, where a User can create a board and grant access to team

members. For this project, a new workspace has been generated, with 3 different boards inside. The main board “Kanban Template SAP Monitoring”, the “organization” board, and the “Sprint example” board, see Figure 11.

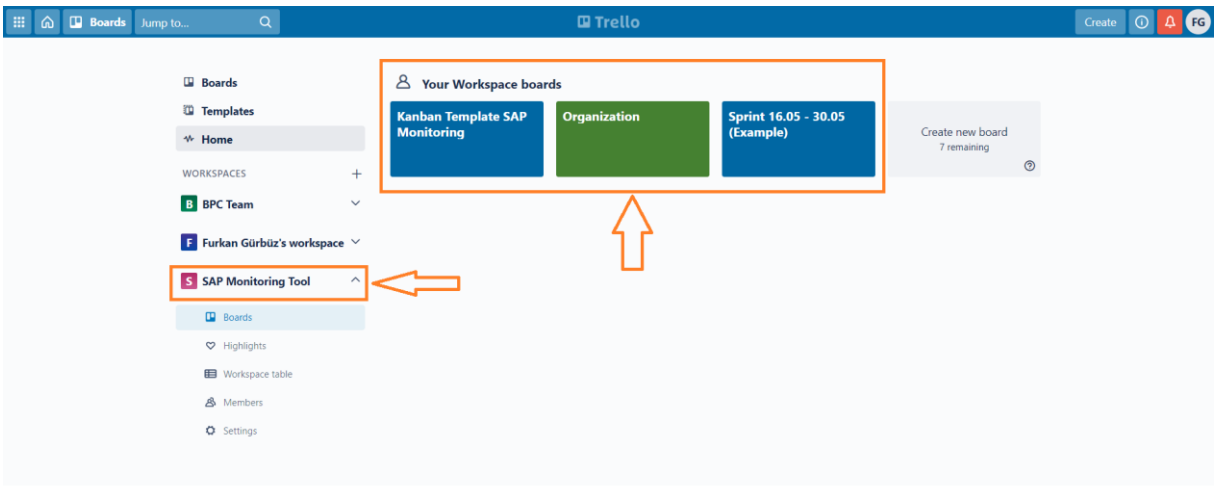


Figure 11: Trello Workspace
Source: own creation

In the Sprint Example Board, the overall definition for specific values has been done, in order to understand the principles within the team. First the overall structure of the board has been defined. There has to be a sprint backlog, which shows the tasks that need to be implemented within the specific sprint timespan. Analogous to the sprint backlog, there should be an “In progress” column that separates the tasks of the actual sprint backlog into the tasks that are currently worked on. Lastly the Done column, where all tasks that are finished should fall in. For the tasks, labels are defined in order to structure the tasks into different categories. These labels are marking the tasks with a specific color that need to be defined with the team members. Tasks with a yellow label are all tasks that fall into the backend programming category, on the other hand the frontend tasks are defined for the category of frontend programming. Additionally the green label, that are optional tasks and red labels, which represent bugs. Furthermore in Trello you have the opportunity to set a deadline, and assign members to the task, as visualized in Figure 12.

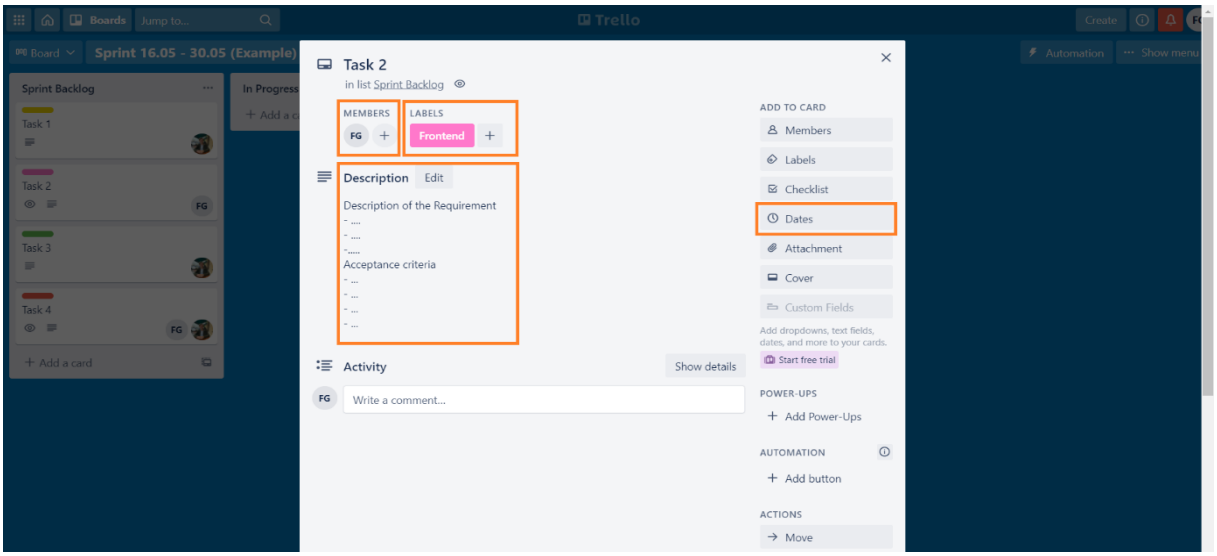


Figure 12: Trello Task description
Source: own creation

Next to the Sprint Example board, there is an Organization board, which visualizes organizational tasks, for example setting up the SAP GUI. The Organization board is structured similarly to the Sprint Example Board. It has the columns Tasks, in Progress, Done, Tutorials for Programming, SAP ACC. The Organization board is visualized in Figure 13.

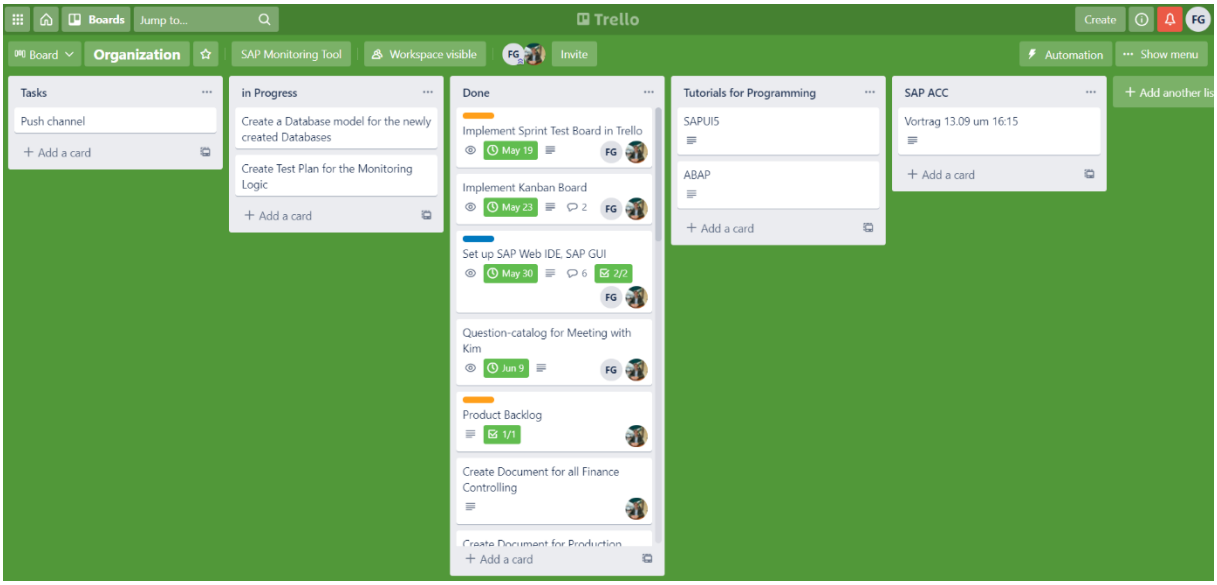


Figure 13: Trello Organization Board
Source: own creation

Next is the main Kanban board, which is declared as Kanban Template SAP Monitoring. This board is used the most with all the elicited requirements that needed to be implemented. It is structured mostly the same as the other board, it has the columns Backlog, Design & Research, To Do, Doing, Code Review, Testing, and Done. The 'Backlog' column is created for all the requirements that need to be implemented. Therefore all the definitions from the previous section need to be stated, which are the label, and with a short description. We did not assign the requirement to a team member or set a deadline for these items in the 'Backlog' because the deadline and assignment are defined in the 'To Do' column. This 'Backlog' is exactly like the Product Backlog of the Scrum methodology. The Next column - which is the 'Design &

Research' column - are requirements, or actions that need to be done within literature research, in order to gain information about a specific topic, instead of programming. An example hereby was the completion of the Excel table, where all items are declared with different weighting and Item Scores. Next, the 'To Do' column. This is actually the column, where the requirements from the 'Backlog' need to be implemented within a given timeframe. Here we additionally set a team member to the requirement, in order to have clear and separate tasks. Furthermore, for the purpose of gaining transparency within the implementation process, the column 'Doing' is created. Here we can drag and drop the requirements from 'To Do' into the 'Doing' column. This helps the team members to comprehend what the other members actively work on in real-time, so we can decrease the chances of a merge conflict within GitHub - more on that in the next sections - and increase the transparency within the development process. The 'Code Review' column was created for improvements in code efficiency, and readability. This process was intended to be done after the implementation of the requirements from 'To Do'. 'Testing' was a more important column, where all functionalities needed to be tested. Therefore different use cases and scenarios have been created. In addition to the short description, a checkbox is added, where the functionalities of the requirements can be defined, and checked if the results are satisfying. This helped us to see whether the requirement has the functionality as desired, and to create a small differentiation within the requirement, and increase the readability of the column. Lastly, the 'Done' column, where basically all requirements that have been successfully implemented are listed. In the following Figure, the mainboard is illustrated.

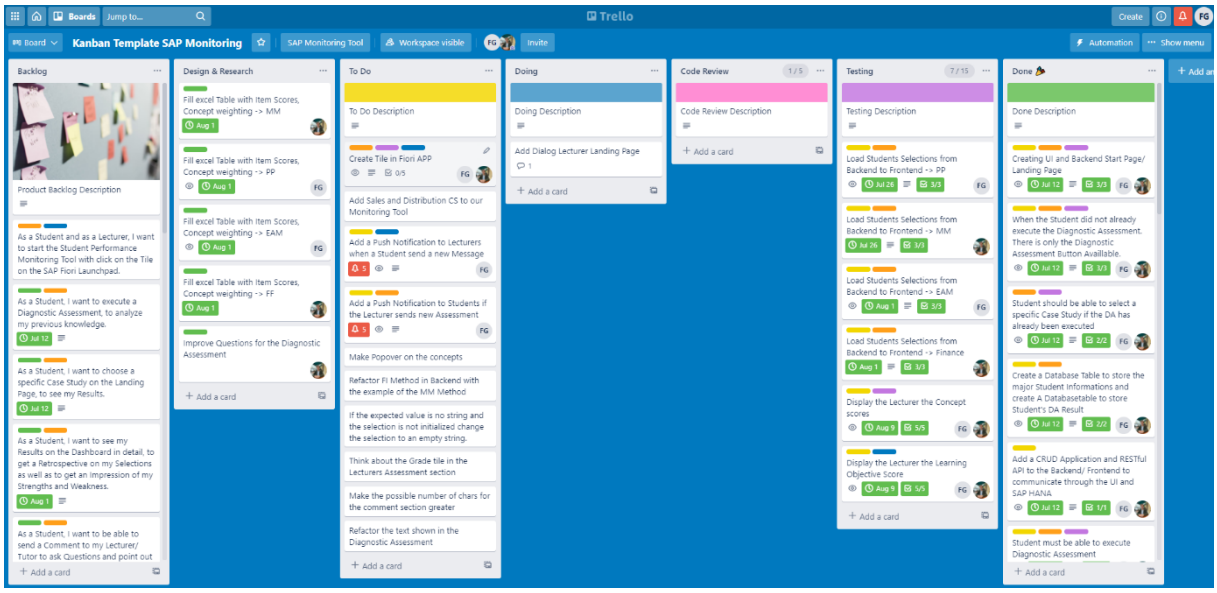


Figure 14: Trello Main Kanban Board
Source: own creation

Since the SAP UCC uses SAP-based technologies, the Monitoring Tool is also based on the SAP implementation conditions. This has the SAP S/4 Hana system, which directly uses the Hana database efficiently. The database has the in-memory technology that keeps and holds the operatively used data not within the traditional electromagnetic hard disks, instead of within the main storage (DRAM). This technology increases the response and processing time up to five times (Gerard & Katz, 2017). For the backend implementation process, the ABAP programming language of SAP has been used, which is connected with the SAP Gateway Service to the frontend. The front end is implemented within the SAP UI5 framework. The overall functionality is adapted from the REST API, which brings the functions GET, POST,

PUT, DELETE (Zhou, Li, Luo, & Chou, 2014). The overall communication between the frontend, backend, and database within the system is illustrated in Figure 15.

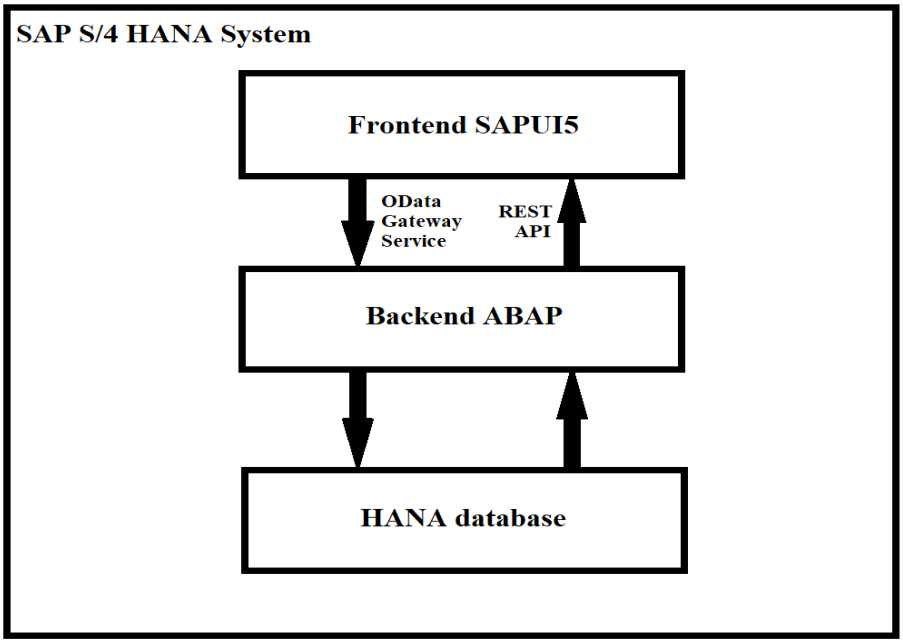


Figure 15: SAP S/4 HANA System
Source: own creation

4.2 Architecture and Data Model

In this chapter, I will present the information, which is mandatory for the software engineering process before implementation. This includes the Architecture and a UML data model of the monitoring tool.

The conceptual UML data model shows the overall structure of the monitoring tool. Therefore we created the main package inside the transaction SE80. The main package is called ZBA_GBIMONITORING, which is the first package in the hierarchy. Inside this package, domains and data types are created, which are needed for the package and sub-packages. Then we defined 2 sub-packages, the ZBA_CASESTUDIES, and the ZBA_DIAGNOSTIC_ASSESSMENT. The creation of sub-packages increases the overall readability and usability. The ZBA_CASESTUDIES package has 4 more packages inside, which represent the actual case studies from the SAP UCC. The ZBA_CS_EAM, ZBA_CS_PP, ZBA_CS_MM, and ZBA_CS_FI. In order to keep the UML data model simple and readable, I only added the concept tables for the Enterprise Asset Management case study, but the methodology is analogous to the other case studies. The concept tables are equivalent to the tasks inside a case study. EAM has 4 tasks inside the case study, which are ‘Create Material Master Data’, ‘Create Activity Type/Work Center/Functional Location’, ‘Create Malfunction Report/Maintenance Order’, and lastly ‘Completion Confirmation/Review of Maintenance Order’. These tasks are defined as ‘CONCEPT1_EAM’, ‘CONCEPT2_EAM’, ‘CONCEPT3_EAM’, and ‘CONCEPT4_EAM’, and assigned in the same order to the corresponding task. Inside the Concepts, we declared two key values, which are always the MANDT and the STUDENT_ID. Next to the key values, we assigned normal values, with ‘ITEM_X’, ‘SELECTION_X’, ‘SCORE_X’, and ‘FEEDBACK_X’. It is important to note that these four values can be interpreted as one small package because these values are the subtasks

inside the whole task and are dependent on each other. The item is the expected solution of the subtask, whereas the selection is the input of the student. Next, the score value saves the reached score after a comparison with the item and selection. Lastly, the feedback, which will be more general feedback on the score and input values. This methodology iterates through the whole tasks and case study, which lead approximately to an item number 20-25, which will not be displayed inside the data model. It is important to mention that the types of 'ITEM_X' in the conceptual UML diagram are dummy values because the purpose of the diagram is to showcase the overall concept and architecture. Additionally, we created two other tables inside the ZBA_CS_EAM package in fact the 'ZBA_EAM_CO_SCORE' and the 'ZBA_EAM_LO_SCORE'. The 'ZBA_EAM_CO_SCORE' table saves the total score items from each concept, that's why the table also has 4 score values, which is analogous to the 4 existing concepts. The 'ZBA_EAM_LO_SCORE' saves the total scores of the concepts with its own mathematical methodology, which will be explained in the next chapter. It has 5 lo score values, because we defined the limit for one case study to maximum 5 executions. Both tables also have the 'MANDT' and the 'STUDENT_ID' as key values. Lastly, inside the 'ZBA_CASESTUDIES' package, a package named 'ZBA_MESSAGES' has been created. Inside this package, the tables 'ZBA_LEC_TP_EAM', 'ZBA_LEC_LP_EAM', 'ZBA_LEC_AP_EAM', and 'ZBA_STUDENT_COM' are inserted. All tables expect for the 'ZBA_STUDENT_COM' save the comments from the lecturer to the student. Therefore different comments from the lecturer can be written to the student, which are comments on the task performance ('ZBA_LEC_TP_EAM'), on the learning progress ('ZBA_LEC_LP_EAM'), and lastly on the activity, behavior and motivation ('ZBA_LEC_AP_EAM'). All tables include the 'MANDT' and 'STUDENT_ID' as key values, and an additional normal value 'INPUT_COM', where the actual comment is saved. Furthermore, the student can also write a comment to the lecturer, which is saved in the 'ZBA_STUDENT_COM' table. The table has the 'MANDT' and 'COM_ID' as key values, and additionally the normal values 'STUDENT_ID' and 'INPUT_COM'. The other sub-package is the diagnostic assessment package, which has the 'ZBA_DA_ANSWERS' and 'ZBA_DA_SOLUTION' tables. The diagnostic assessment asks the students five simple yes/no questions, which are:

- "I have already worked actively with SAP S4/HANA and know exactly how to use it. For example in a working student job or an internship"

- "I am aware of the impact to which the structure of the Materials Management, Production Planning, EAM, and Finance business units will change as a result of GBI's digital transformation"

- "I am taking this course because I want to train to work with SAP S/4HANA and to understand how it works and how it is structured. The goal is to get an expert in working with SAP S/4HANA"

- "Ideally, I would like to try to go through each case study at least three times, so that the knowledge I have gained in the context of digital transformation and SAP ERP will be improved"

- "I will consciously use the Student Performance Monitoring Tool to self-reflect my results and to see in the retrospective which concepts I am still struggling with."

The input answers of the student are saved in the 'ZBA_DA_ANSWERS' table, whereas the overall score that is calculated with a mathematical model is saved in the

'ZBA_DA_SOLUTION' table. The conceptual UML data model is visualized in the following figure.

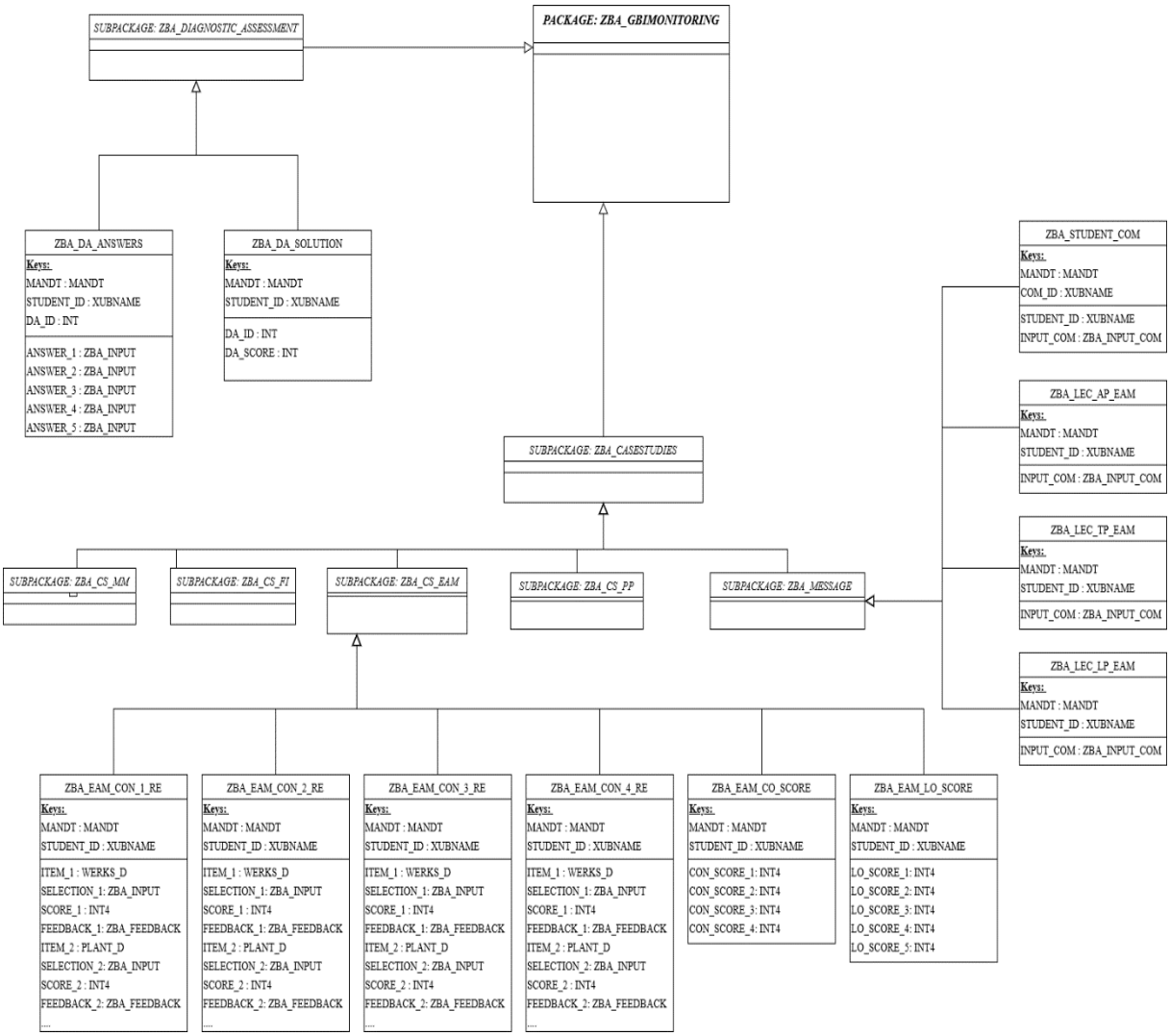


Figure 16: Conceptual UML data model
Source: own creation

Next to the data model, a component diagram has been created and visualized in Figure 17. The frontend and backend consist of two components, which are in the frontend the Student's and Lecturer's View, and in the backend the request processing and algorithm. The associated view in the frontend will be managed by the implemented access control, which queries and identifies the currently logged-in user. Then the user gets redirected to the associated view. Both views have access to the data via the backend. The connection between frontend and backend is carried out with the REST API. Then the requests are progressed, which in addition triggers the function/algorithm that is implemented in the backend and gives back the requested data.

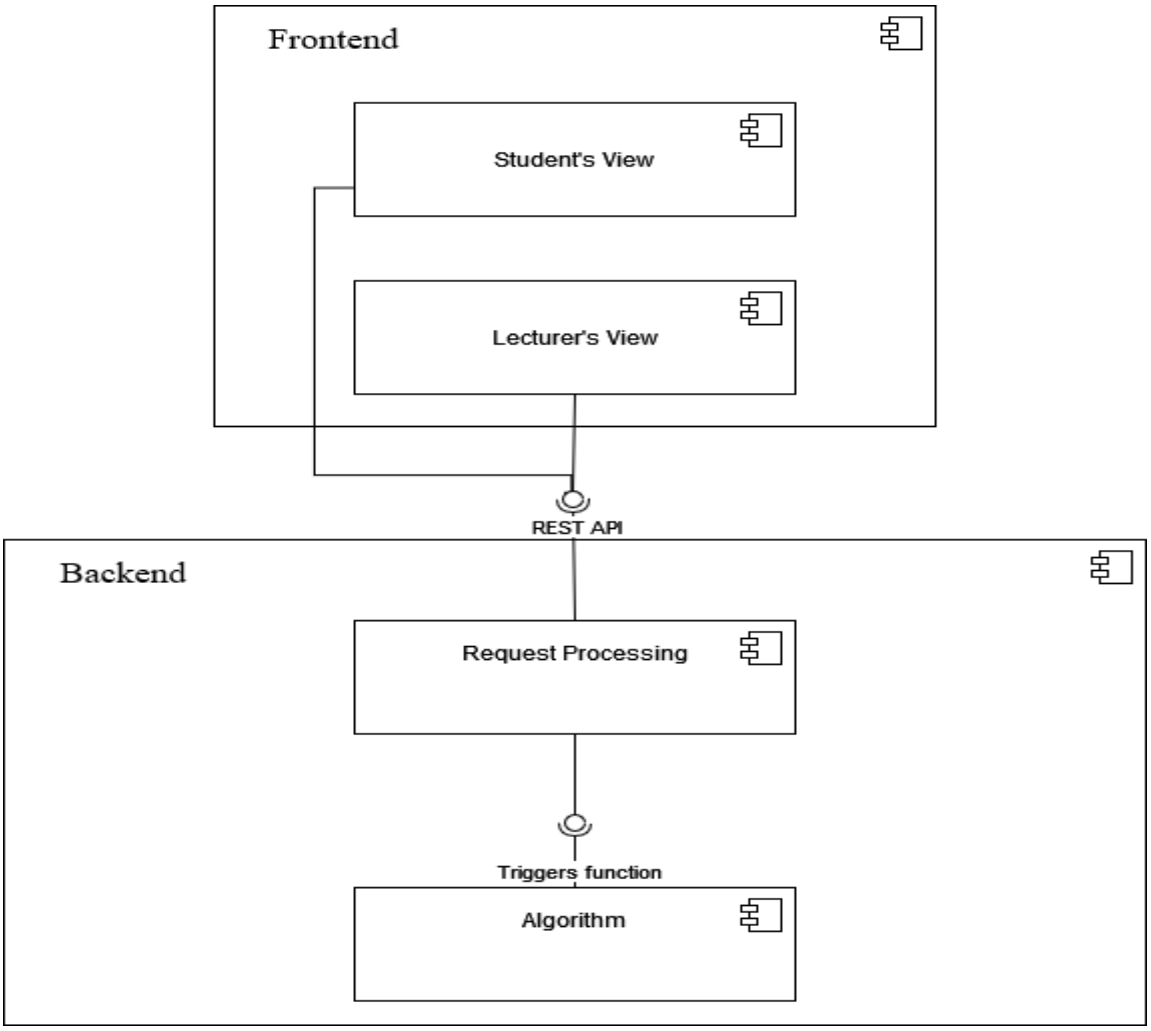


Figure 17: Component diagram
Source: own creation

In this sequence, the communication and overall concept will be explained. Therefore a sequence diagram has been created and visualized in Figure 18. As mention in the previous section, the monitoring tool has two components in the frontend for the associated user. That's why two different actors are included, which are the Lecturer and the Student. In between the Application is visualized. In order to be able to follow the overall concept, the Lecturer has a method identification number starting with the digit 1, whereas the Student method calls have the identification number 2. The student can log in directly by entering his login details, whereupon the system redirects the student to the associated page. Hereby it is important to note that the login process is identical for the lecturer. Now the student has the opportunity to complete the diagnostic assessment, which directly has the consequence, that the lecturer is now able to monitor the student. The reason for this is that the diagnostic assessment also functions as an entry into the monitoring system. This is done via the REST creation method, which creates and pushes the user into the database with the completion of the diagnostic assessment. In the sequence diagram, I differentiated between two loops, in order to make it clear that the monitoring process of the student and lecturer are not dependent on each other. This basically means that the student can always iterate through the monitoring process, without the lecturer having to monitor the student. First, the loop of the lecturer starts with the selection of the student. The data hereby is displayed with the read method of the REST API. Then the overall results of the student within the case study is being calculated in the backend and saved

in the database, and then pushed into the frontend with the read method of REST. The calculated data is showcased with different diagrams, graphs, and other statistics. The next functionality of the lecturer is the comment function. Here the lecturer can write a comment which is pushed from the frontend to the backend with the create method. The comment basically creates an entry into the table with the important values like the student id, the actual comment, and the category. The comment is visualized in the student's frontend on the lecturer's assessment page, where the comment is being pulled via the read method from the REST API. Now the student's loop will be explained. Therefore the student can select a case study, which is hardcoded in the frontend because only 4 case studies exist in the scope of this thesis. After the selection of the case study, the corresponding data will be showcased with the read method of the REST API. The data is shown as a table-like structure, where the following columns are created: Item, Your Selection, Score, and Feedback. The score values and feedbacks are generated and calculated within a method, that is being triggered from the frontend, and is implemented in the backend. The comment functionality is basically analogous to the lecturer comment functionality, the only difference is, that the student can not categorize his/her comment for the lecturer. Hereby the creation method of the REST API saves and pushes the comment that is written by the user in the frontend, into the backend.

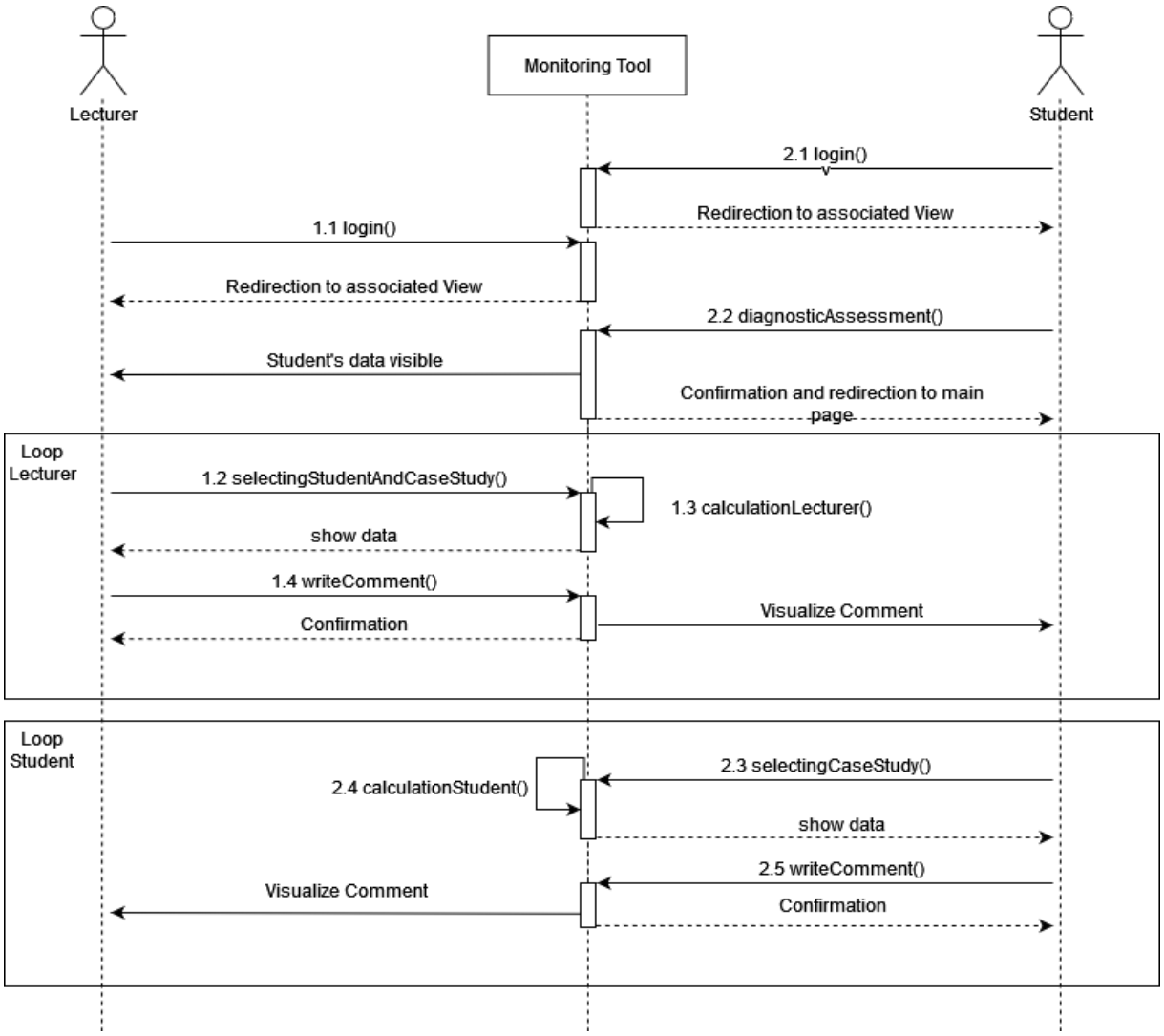


Figure 18: Sequence diagram
Source: own creation

4.3 Implementation of the Requirements

In this section, the implementation of the requirements that were defined in the previous sections will be explained. Therefore I will showcase screenshots of the associated requirements that have been implemented. Because the development process has been divided between me and Mr. Mustroph, I will only visualize and explain the requirements, which I have been implemented.

4.3.1 Organizational Requirements

As mentioned in the previous sections, the monitoring tool has two different views. On the one hand the lecturer, and on the other hand the student view. The redirection to the associated page is handled with the login details. The SAP UCC has been defined as a methodology for the UserIds for the student. The Id consists of three letterals at the beginning, which is 'GBS', and in addition to the letterals, the id has a 3 digit number. These two components are merged with a '-' in between. These combinations make the id unique, an example for a student id would be 'GBS-995'. For the admins, there does not consist of a specific definition of the ids. That's why we programmed into the onInit function of the main page, an if statement, where the user id is controlled about the methodology of the student id. This basically means, that the if statement control if the id of the currently logged-in user starts with 'GBS'. After this control, the tool knows, if the current user is a student or a lecturer. Next, the user gets redirected to the associated page. The Lecturer is redirected to the page, which is visualized in Figure 19. It is also important to note that this page is actually also the main page for the lecturer. There is also the recognition in the center of the page, where the lecturer can see, that he/she is successfully redirected on the lecturer page.

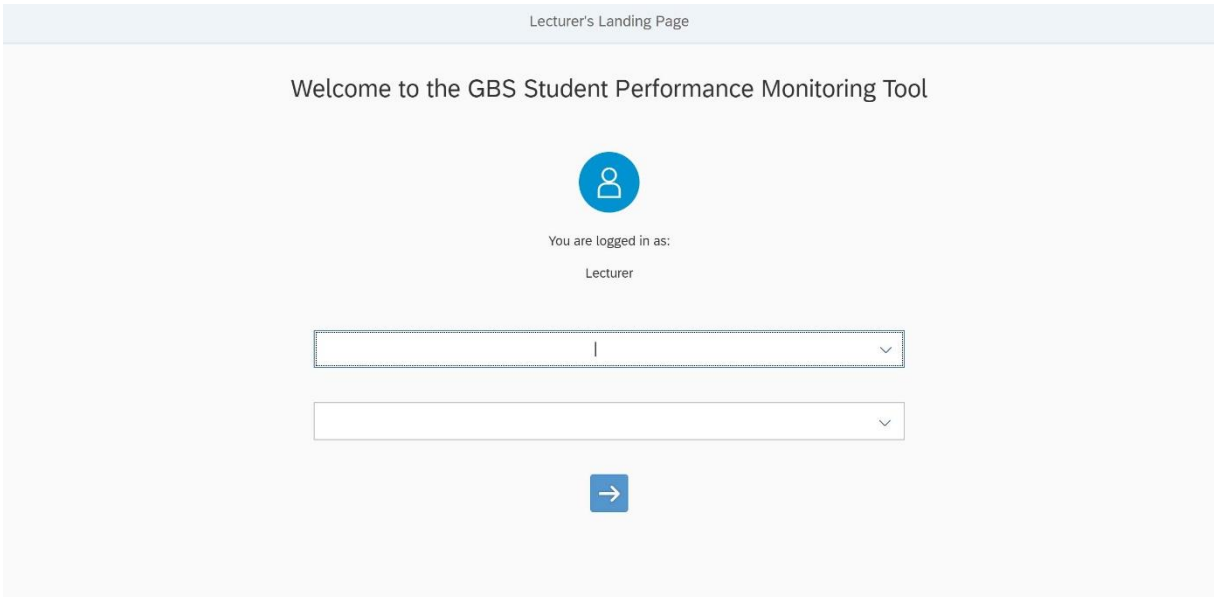


Figure 19: Lecturer Landing Page
Source: own creation

Now the same process also goes for the student, where the system redirects the Student on his/her pre-landing page, which has different functionalities, for example, the diagnostic assessment. The landing page of the student is visualized in Figure 20. There you can see the two buttons in the center. One button sends the student directly to the diagnostic assessment page, and the other button directs the student to the main landing page, where he/she can select the desired case study from the dropbox. In addition to the buttons, the student also has the confirmation of a successful redirection, which is located in the center of the page, where the user id is stated.

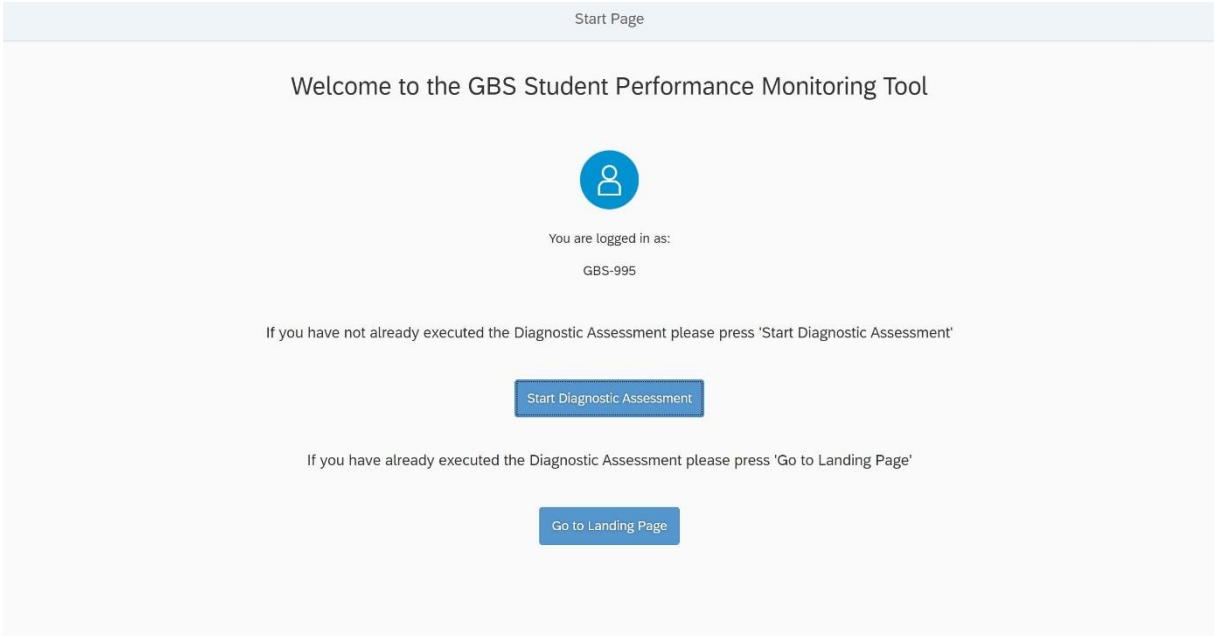


Figure 20: Student Pre-Landing Page
Source: own creation

4.3.2 Requirements for the Set of Action Default User

After getting redirected to the associated page, the student sees the pre-landing page (as in Figure 25). The Diagnostic assessment button sends the student to the diagnostic assessment page, where he/she can answer and submit the questions of the assessment. After the submission, the student gets directed to the main page. This page can also be accessed from the pre-landing page without having to complete the diagnostic assessment. Therefore the user has to press the ‘Go to Landing Page’ button. The landing page is visualized in Figure 21.

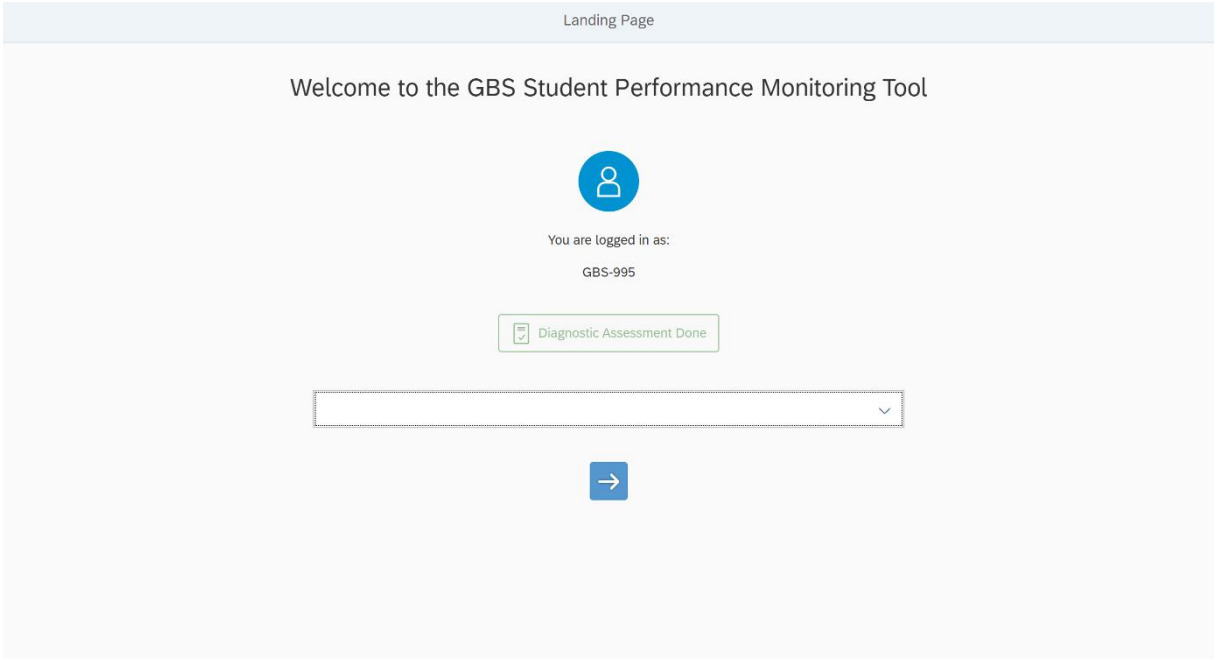


Figure 21: Student Landing Page
Source: own creation

On the landing page, the student can see again the login details in the center, and in addition, the diagnostic assessment is stated in a green box as “Diagnostic assessment done”. Furthermore, the student can select the desired case study from the dropbox, where the case studies are hardcoded into the dropbox because in the scope of this thesis only 4 case studies had to be implemented. Every item in the dropbox has a unique id and the selection of the student can be fetched from the controller with the selectedKey() method, where the selected key is fetched. Afterward when the user clicks on the arrow button, which basically confirms the selection, an if statement controls what element has been selected and redirects the user to the desired case study page. Next after selecting and confirming the desired case study, the student gets directed to the requested page. Here the student can see his/her entries that he/she has done within the case study tasks. Hereby the feedback and score values are automatically calculated, and if the student makes a wrong entry the visual output is also adapted to the score, which will be marked in red. The page has a header, which confirms the user, that he/she is on the desired case study page. In addition, the user can select the different concepts/tasks inside the case study. The implementation and separation of the different concepts increase the readability, and the overall structure of the page. Furthermore, a small box with an explanation about the concept task has been added. The mentioned scores and feedbacks are represented in a table-like structure, which increases the readability. Additionally, the student can write a comment for the lecturer on the specific case study, but more on that later. In the footer, the student can click on the ‘Lecturer’s Assessment button’, where the student sees comments and the assessment of the lecturer.

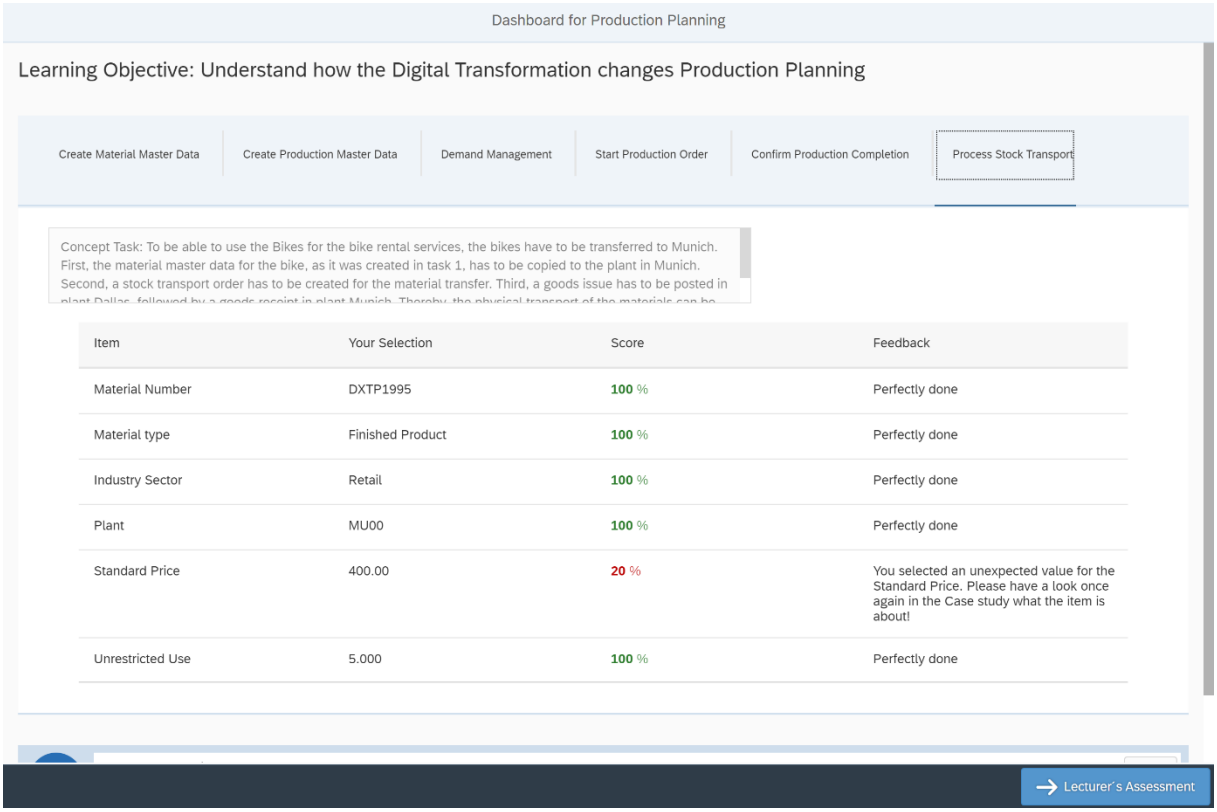


Figure 22: Student Dashboard for Production Planning
Source: own creation

4.3.3 Requirements for the Set of Action Admin User

In this section, the set of actions of the lecturer will be explained and visualized with screenshots. Therefore the first step bevor this section is the login process, so after the lecturer logs in to the tool, the lecturer gets redirected to the Lecturer landing page. The Lecturer landing page is similar to the student main landing page but has the opportunity to select individual students, which the lecturer wants to monitor. As you can see in Figure 24, the lecturer has two dropbox opportunities. The upper dropbox functions for the selection of the student. To Achieve this, the students are saved in the database after conducting the diagnostic assessment, which functions as registration into the monitoring tool. Within the dropbox, the read method from the CRUD API gets the data from the backend to the frontend. The lower dropbox is identical to the dropbox from the student, where the case studies are listed and selectable.

After selecting the desired student and case study, the lecturer gets to the lecturer dashboard page, where he/she can see the overall scores, achievements, and activities for the student. In order to showcase the dashboard, the student 'GBS-995' and the case study 'Production Planning' has been selected. The following figures are screenshots of the lecturer dashboard page. Figure 28 showcases the first statistic that the lecturer can see. Hereby it is also important to note that a small confirmation of the selected user is visualized in the marked upper left spot. The statistic showcases the task performance results, which are basically the achieved scores inside the case study Production Planning. As mentioned before, the concepts represent the tasks inside the case study. In this demo, you can see that the student GBS-995 conducted the case study with high score values.

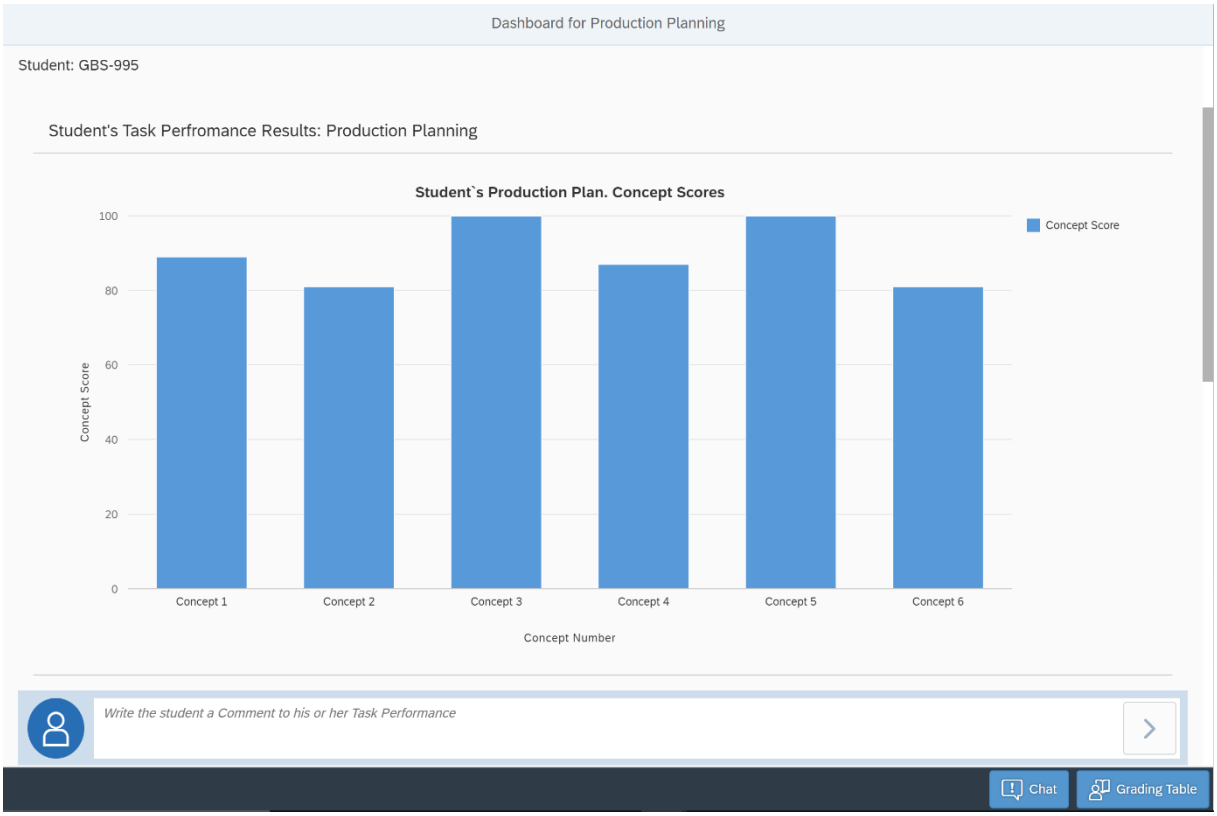


Figure 23: Lecturer Dashboard for Production Planning 1
Source: own creation

In the next Figure, the statistics of the learning curve for the case study are showcased. The Lecturer can see the student's overall learning objective score, which is calculated with mathematical formulas inside the algorithm. The learning curve is a showcase with the number of executions in the x-axis and the learning objective score in the y-axis. The number of executions is limited to 5 times, which is approved and accepted by the SAP UCC because according to the experience of the lecturers the students do not conduct the same case study for 5 times. If a student does the executions more than 5 times, this has the consequence, that the scores and achievements of the lastly conducted execution will not be saved in the database and can not be visualized.

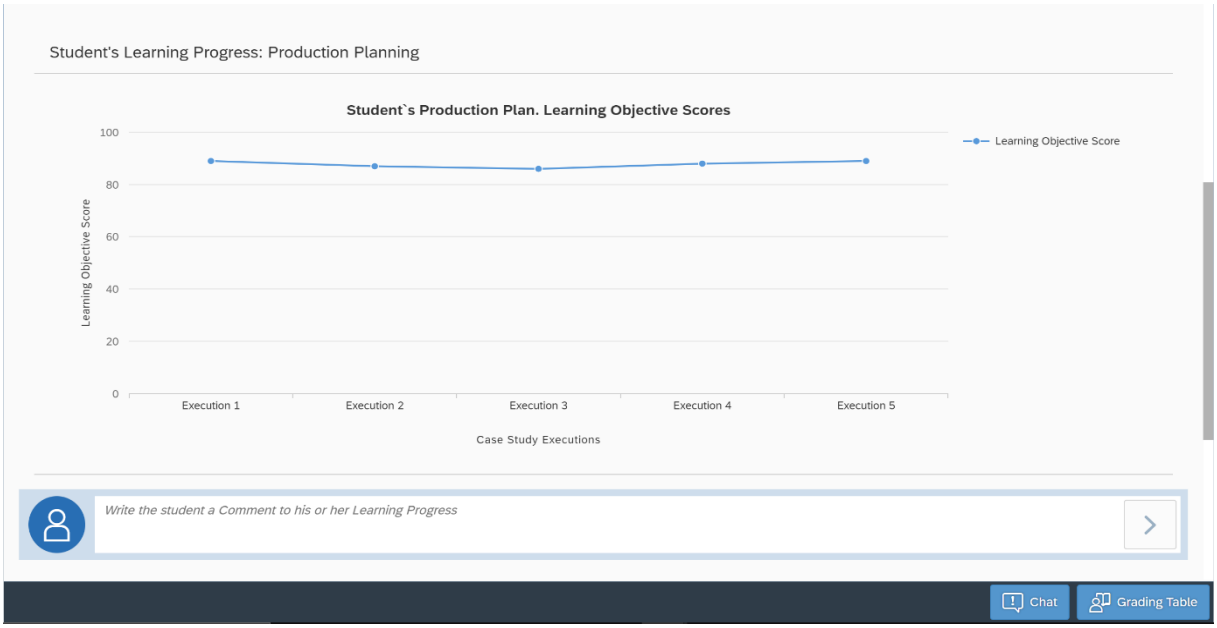


Figure 24: Dashboard for Production Planning 2
Source: own creation

An additional statistic is showcased, which has the information about the student's activity and diagnostic assessment result, compared to the whole course. The student's activity is measured with the number of executions of the specific case study. The data is showcased inside an icon box, which is not clickable, whereas the icon box for the diagnostic assessment is clickable, which is also visualized in the icon box with the statement "Click to see all Details!", which is visualized in figure 25.

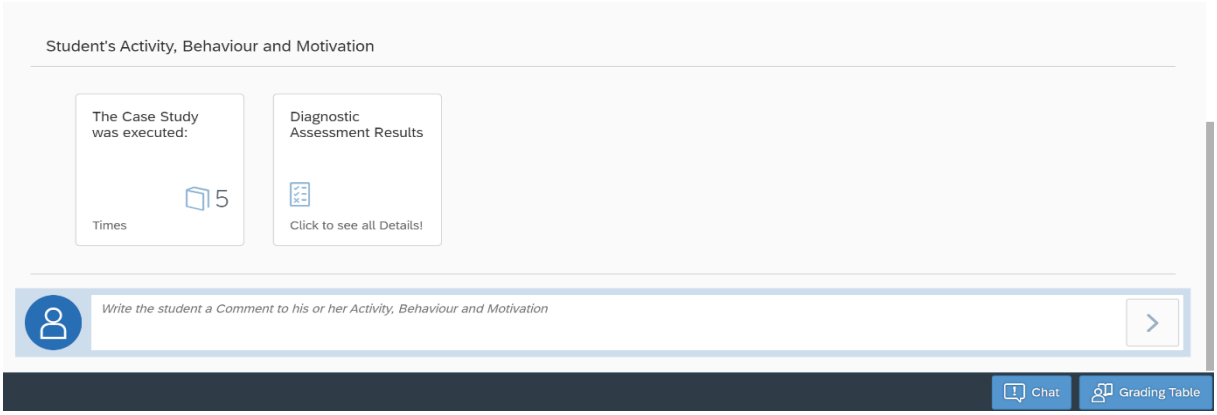


Figure 25: Dashboard for Production Planning 3
Source: own creation

After the Lecturer clicks on the Diagnostic Assessment Results Icon Box, he/she gets redirected to the page visualized in figure 26. In the upper part of the page, the information box includes an explanation of the overall methodology of the page. The Lecturer can see here the results of the student from the diagnostic assessment, compared with the average score of the course.

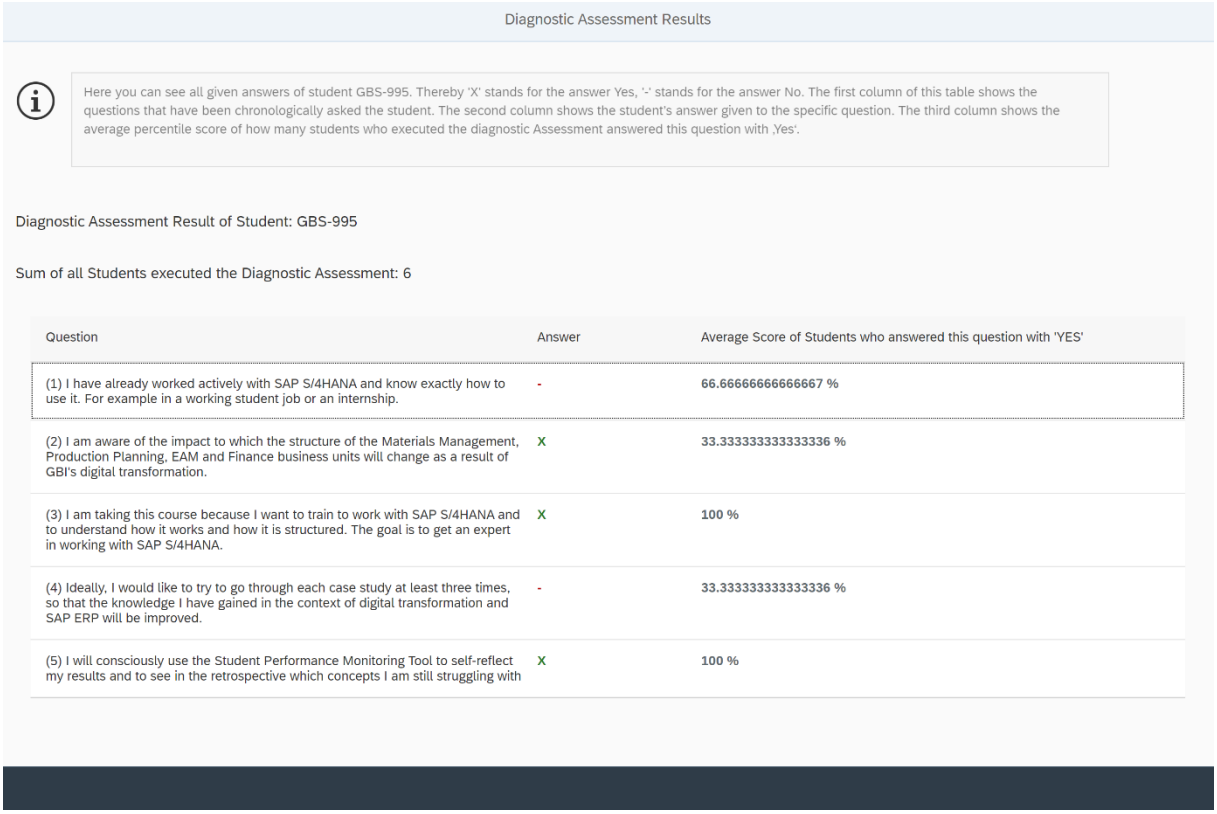


Figure 26: Diagnostic Assessment Results Page
Source: own creation

4.3.4 Requirements for the Comment Section

The comment functionality has been already mentioned and noticed in some of the figures from the previous section. The comment functionality is motivated by the increase of the interaction between student and lecturer, which is fundamental in an e-learning environment. First I will talk about the comment functionality for the student, which is kept simple. As visualized in figure 27, the page is the same as in figure 22, where the student can conduct a self-assessment on the Student Dashboard Page. Here you can see a small comment box with the text “Here you can write a Comment to your Lecturer”. After writing the comment and clicking on the arrow button, the comment gets saved into the database with the sender and receiver information. If the process was successful, then the system displays a success message. This Comment does not have any specific topic or specialized category, but it is within the scope of the specific case study, that has been selected in the previous sequences.

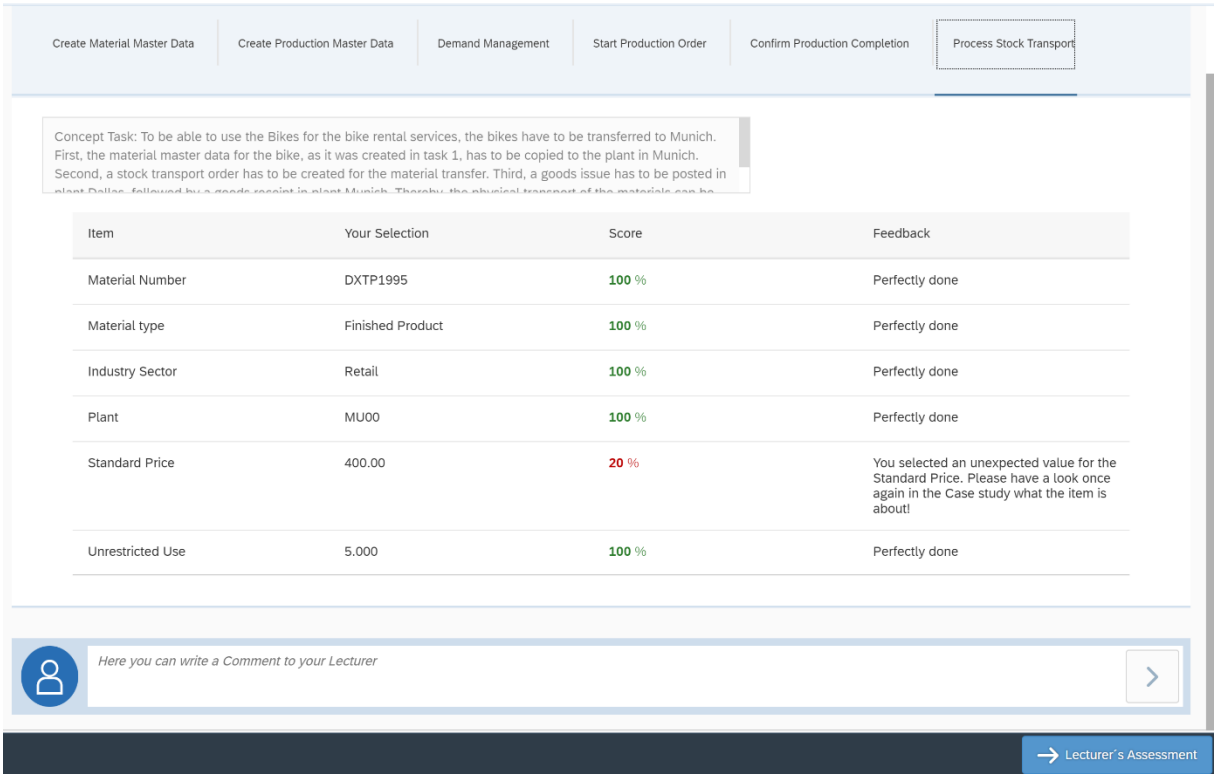


Figure 27: Comment functionality Student
Source: own creation

The Lecturer Comment functionality differs in the categorization of the lecturer's comments directed to the student. As stated in the previous section (4.3.3) the lecturer has different statistics of the student. Therefore the lecturer should be able to comment on each statistical value. In figure 23 the lecturer assesses the student performance of the concepts within the case study. Hereby a comment box exists with the text "Write the student a Comment on his or her Task Performance ". This comment will then be saved in the database and showcased on the student's Lecturers Assessment Page. Additionally, the statistics visualized in figure 24, which showcases the learning objective scores, have the comment functionality, so the lecturer can comment on the students learning curve. Lastly, the activity of the student, which is visualized in figure 25, can also be commented on by the lecturer.

The Lecturer's Assessment button is very important for the student because this is where the constructive feedback of the lecturer is given. As mentioned before, the lecturer can comment on different statistics of the student. These comments are saved in the database and showcased on this page. Hereby the comments are also structured in their category, so the student can easily identify on which topic the lecturer is commenting. The grade icon box is also non-clickable and represents the overall score and feedback of the lecturer. The page is visualized in figure 28.

Next to the student's assessment page, there has been created a page for the Lecturer, which showcases all comments from the student. Hereby the Lecturer has to click on the "Chat" button, which redirects the Lecturer to the Chat Page, where all comments from the students are gathered and showcased. Here is no filter functionality or any other type of categorization, which will be discussed and mentioned in the last chapter "Conclusion and Outlook". The page for the lecturer is visualized in figure 29.

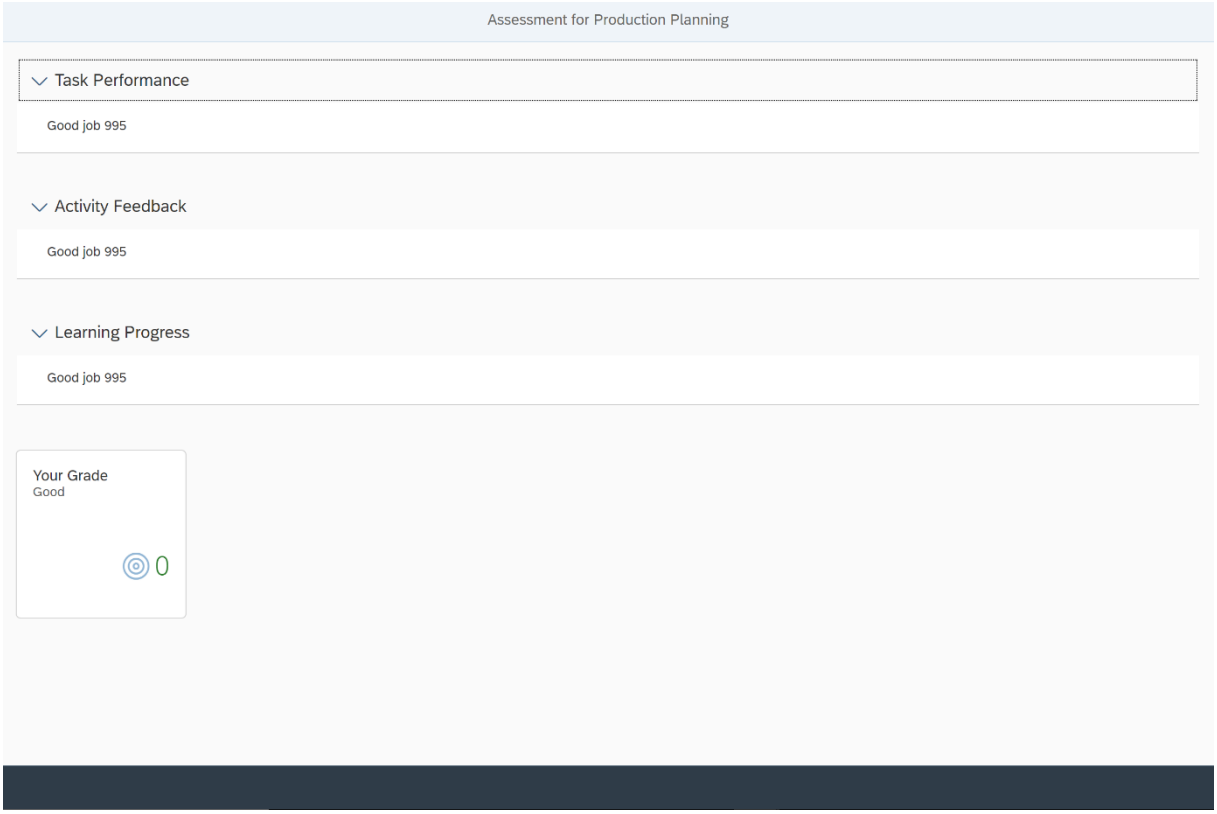


Figure 28: Lecturer's Assessment Page
Source: own creation

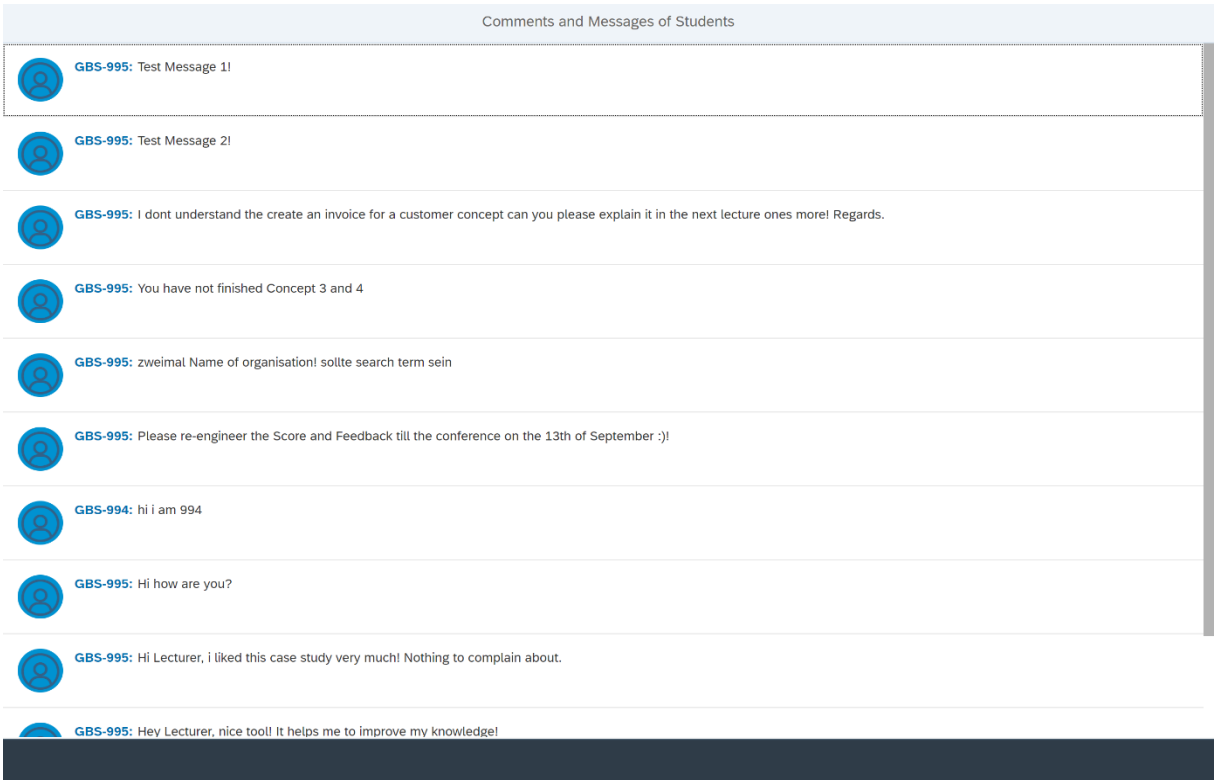


Figure 29: Comments and Messages of Students Page
Source: own creation

4.3.5 Requirements for the Feedback

The feedback is a generated value in the algorithm and mathematical model within the didactical side of the prototype. The feedback is based on different values. Because the overall algorithm is explained within the thesis of Mr. Mustroph, but in this section, I will shortly give a brief view of the generation of the feedback, which is part of the algorithm. In order to get a better understand, a conceptual model has been created in figure 30. Here you can see the if statement, which checks, whether the input of the student, is exactly the same as the expected solution value. If this is the case the feedback "Perfectly Done!" is saved. In the other case, the input is controlled, whether the input has a value or not. Because of simplicity reasons, the wrong input is broken into this if statement. If the if statement is wrong, then the feedback is "You selected an unexpected value for …!". On the other side, the feedback "You did not initialize a …!" is generated. In figure 27 is an example of the generated feedback on the student's dashboard page!

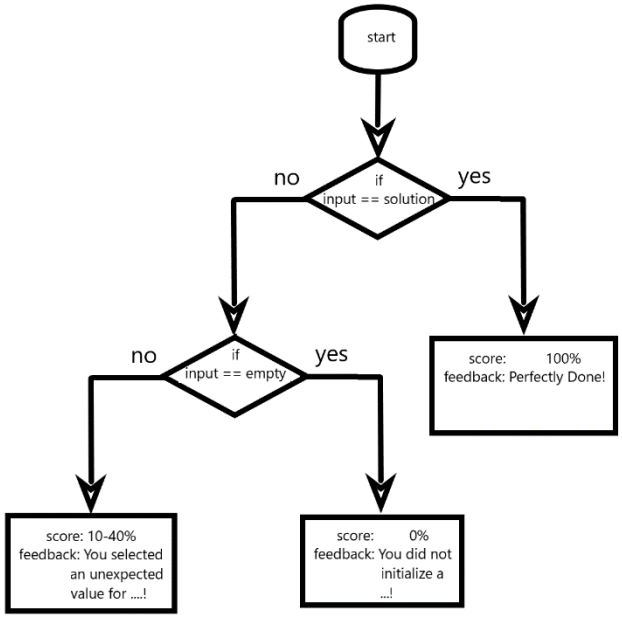


Figure 30: Conceptual model for feedback generation
Source: own creation

4.3.6 Non-functional Requirements

The Non-functional requirements are stated in the table from the previous section, which include the NF01 Accessibility, NF02 Response Time, NF03 Availability, NF04 Language, NF05 Event of Failure, NF06 Design, and lastly NF07 Usability. The important requirements will be discussed and explained in this section.

First of all, I will talk about the usability of the prototype. Usability is defined as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" (Kaya, Ozturk, & Gumussoy, 2019). This requirement is one of the most important requirements next to the functionality of this specific tool. In order to create a usable tool, a quick click-flow and pages diagram has been

created in figure 31. There you can see the overall page structure of the prototype, which directly connects with the usability requirements. The main page is basically the student's main page, where the access control redirects the currently logged-in user if the user is a lecturer! The lecturer gets redirected to the lecturer landing page and has no opportunity to go back to the main page. The main page has two buttons as described before and can go to the diagnostic assessment page and student landing page. Here it was important to have the transitive relation between these three pages in order to increase the click flow. Within 4 clicks, the student can reach every page, which additionally increases the usability of the page. Next to the student, the lecturer also can reach every page within 4 clicks. This showcases the overall usability and click-flow of the prototype.

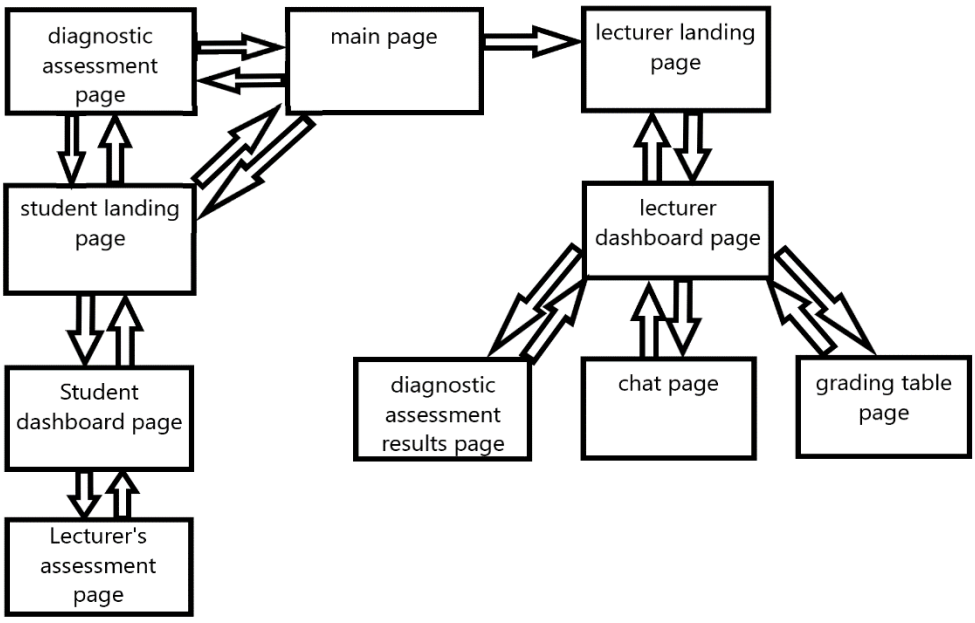


Figure 31: Click-flow and pages diagram
Source: own creation

Now the design requirement will be explained and showcased with a page structure overview. As stated in the previous chapter, the overall design of the monitoring tool should maintain the sap Fiori design, within the sapui5 frontend framework. Therefore a mock-up and overall structure of a page have been showcased in figure 8, which visualizes the overall design idea of the pages. There have been created two mock-ups, on the one hand, the page for the student, and additionally the page for the lecturer. In figure 27, the mock-up page for the student, and in figure 38, the mock-up page for the lecturer is visualized. Both pages have the common overall structure of the design concept of the sap Fiori design (figure 8) with a header, a simple data visualization section, and a footer. In addition, an information and description sequence has been added to the design, in order to increase the usability and understandability of the pages. The student mock-up page showcases the data within a table-like structure, in order to increase the readability, whereas the lecturer can see statistics and result scores about the student. The Comment and page navigation is in the design perspective identical and has no difference.

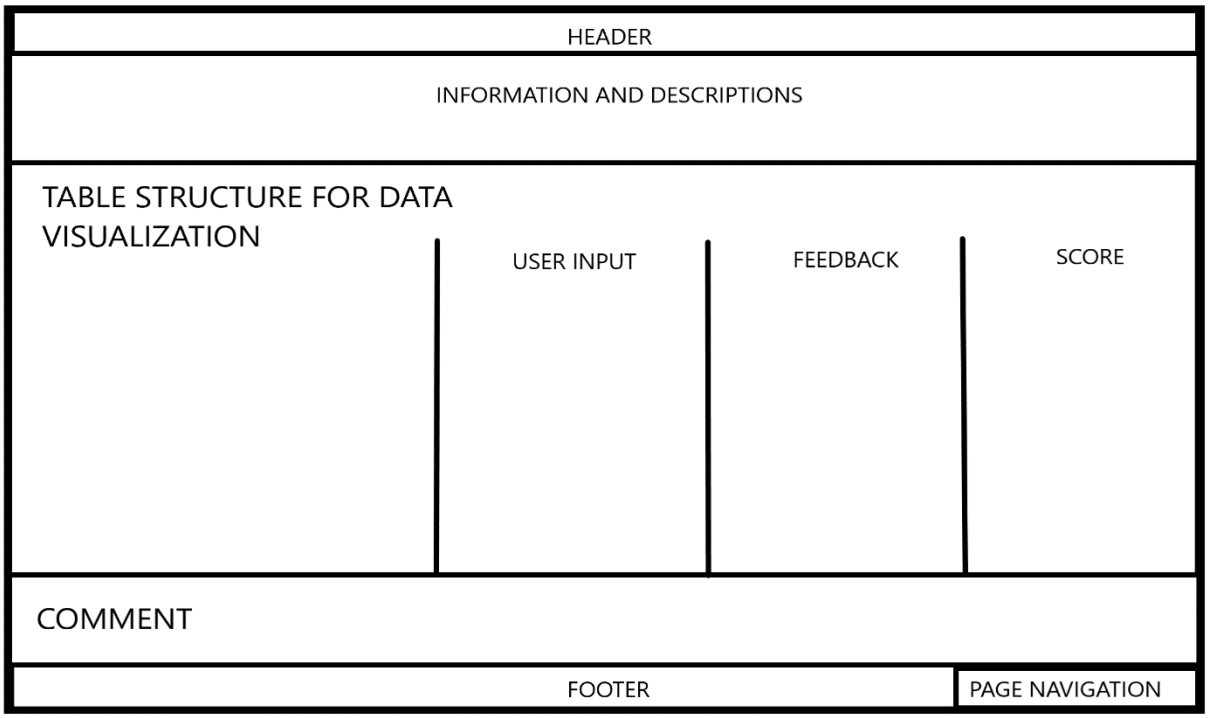


Figure 32: Mock-Up Page for Student
Source: own creation

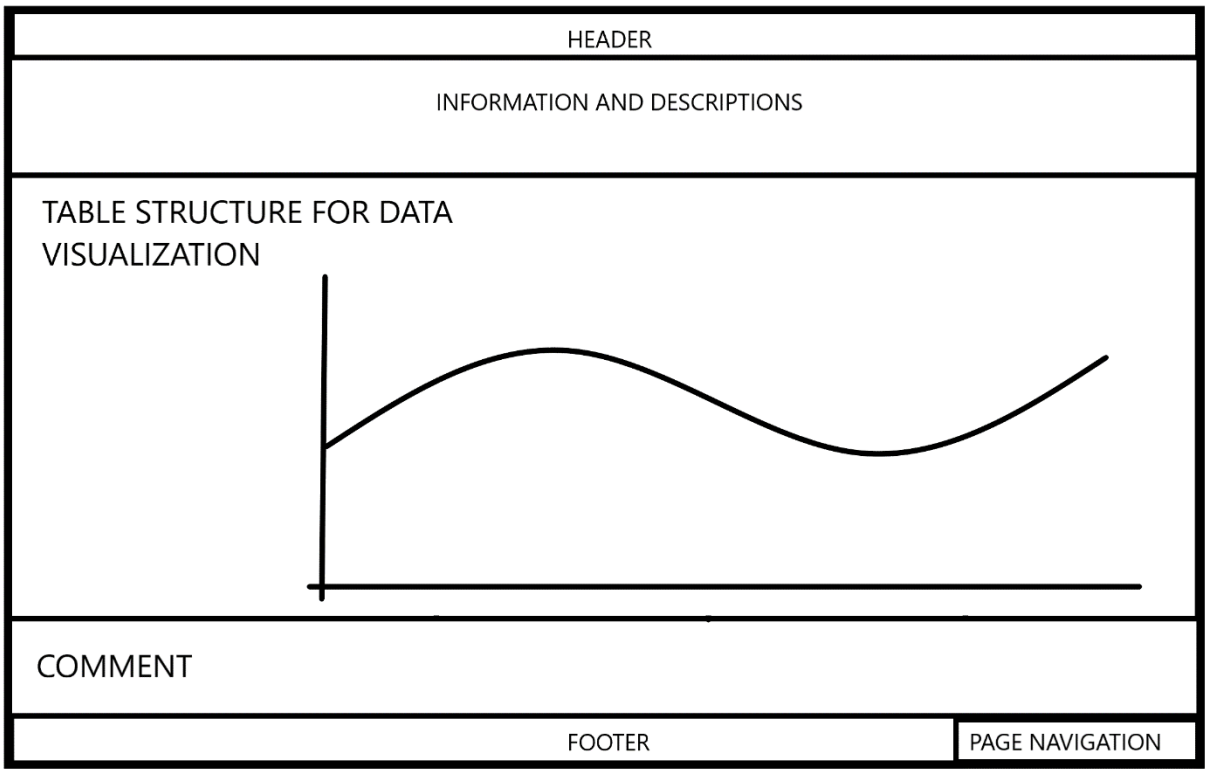


Figure 33: Mock-Up Page for Lecturer
Source: own creation

4.4 Results Evaluation

After the conceptualization of the requirements document in chapter 3, the prototype has been developed based on the SAP S4/HANA system, within the SAPUI5 framework for the frontend development, and the ABAP language in the backend. The data is transferred with the sap gateway service, which uses the REST API. All requirements have been implemented within Sprints, after the SCRUM methodology. The student can monitor and self-assess himself/herself within different case studies in order to improve the overall education. The student can also send comments to the lecturer, which adds value to the interaction between student and lecturer. However, the lecturer can view the statistics and scores of the student, in order to monitor the student performance over a case study. After the implementation process, the developers have conducted penetration testing, where the case studies have been done, and the results are monitored. In the following sections, the monitoring tool will be evaluated with a student group, which will be discussed in the next chapter.

5 Evaluation of the Monitoring Tool

In this chapter, I will talk about the evaluation process and results of the developed monitoring tool. In the first section, I will present my methodology of the evaluation process, give information about the evaluation formula with the asked questions inside this formula. Next, I will discuss and interpret the results from the answered evaluation formula by the students'. It is important to mention, that the whole evaluation process is based on the student's view of the monitoring tool.

5.1 Methodology

As mentioned before, the evaluation process of the monitoring tool is based on the student's perspective. In addition, this thesis has the focuses on the technical requirements of the monitoring tool, which will be evaluated. In order to evaluate the technical requirements, it is important to evaluate the user experience of the tool. The Usability of the tool can be evaluated by the System Usability Scale (SUS). Therefore a questionnaire has been prepared for the students, which is basically based on the SUS evaluation formula. The original SUS formula is visualized in figure 34. The SUS is a Likert scale with a range of a degree of agreement or disagreement inside a 5 point scale (Brooke, 1996). The SUS evaluation contains ten simple and basic questions, which have the main topic about the usability of the system. The overall aim of the evaluation is to understand encountered problems of the students while using the monitoring tool (Kaya et al., 2019). According to Kaya, the SUS evaluation increases the simplicity, and quickness of the evaluation process for the participants and practitioners (2019).

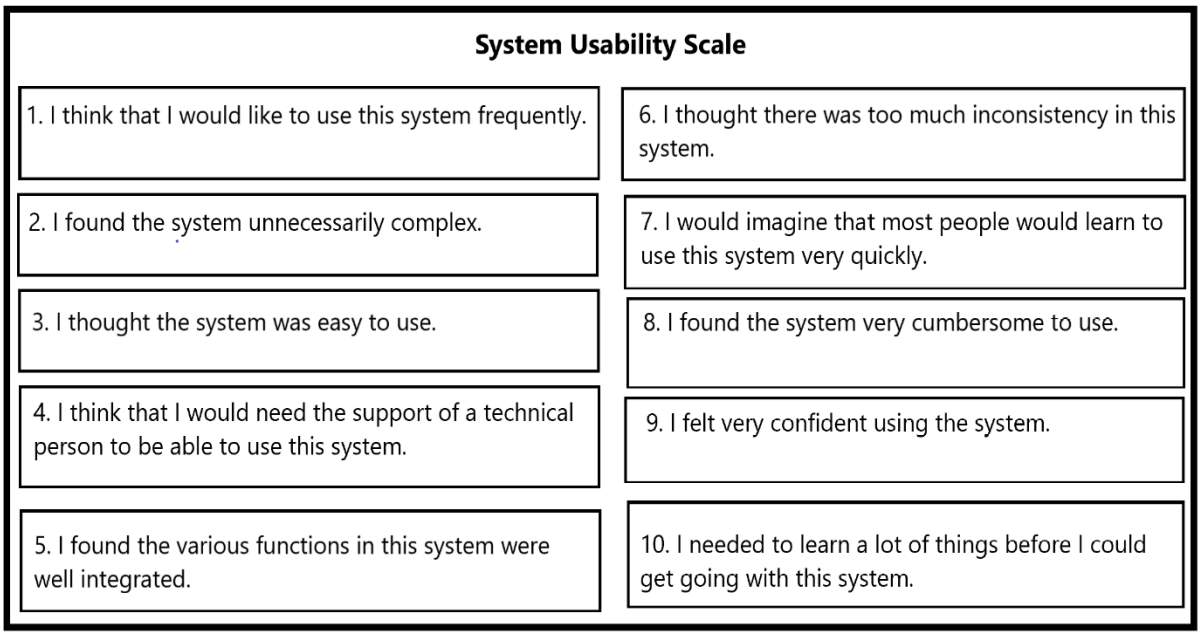


Figure 34: System Usability Scale
Source: own analysis

The SUS has to be tailored for the own needs. Therefore the word "system" inside the sentences has to be changed into "monitoring tool". In addition, the SUS has a defined score calculation, which can be easily calculated, in order to interpret the overall results. For the calculation of the evaluation, the predefined SUS calculation will be done. Therefore the definition of Brooke

for the calculation is: “To calculate the SUS score, first sum the score contributions from each item. Each item’s score contribution will range from 0 to 4. For items 1, 3, 5, 7, and 9 the score contribution is the scale position minus 1. For items 2, 4, 6, 8, and 10 the contribution is 5 minus the scale position. Multiply the sum of scores by 2.5 to obtain the overall value of SU” (1996).

In this section, I will present my evaluation process. As mentioned before the evaluation formula questions are combined with the predefined questions from the SUS methodology and demographic questions like the course of study, the age, and the semester. Among the participants, only students have been taken into consideration without a limitation of the course of study. The goal for the number of participants was a minimum of 10 students. That’s why the evaluation process was restricted because of the overall corona situation. Therefore a virtual method was mandatory for the evaluation process. First I have done a small poll among the evaluation participants, where the date and time of the video conference have been defined. Nevertheless, there was no date, where a majority of the participants could attend. That’s why I decided on another method, where I upload a video about the overall evaluation process and explain every aspect of it. This video has been uploaded into a google drive folder. The next step of the evaluation was the link and login data for the monitoring tool. Therefore I posted the link into google drive, but the login data has to be personally requested, because of privacy policy reasons. Lastly, I uploaded the evaluation formula into the drive, which can be conducted. In the last folder, I uploaded an example of the case study production planning. In figure 35 the structure of the folder “Evaluation” inside the drive is showcased.

Figure 35: Evaluation folder in Google Drive
Source: own creation

In order to make the evaluation process more simple for the participants, the mandatory folders are numbered, so the participant exactly knows what order to open the corresponding folder. As mentioned before, the video explains every aspect of the evaluation process. Therefore I created a PowerPoint presentation for the video, where the motivation, methodology, and the monitoring tool are explained. Any further questions that came up after the video, can be asked in a one-to-one virtual meeting, or via e-mail.

The link for the monitoring tool is posted in the second step after the participant has watched the video and has a better understanding of the overall process. There he/she can click on the link and will be redirected to the login page of the monitoring tool. The login data for the monitoring tool has not been posted into the drive because of privacy policies. That’s why the login data had to be personally requested via message or e-mail. Every participant did get a randomly assigned “GBS-XXX” user.

After the student analyzes the monitoring tool and got himself/herself an opinion on the tool, the participant can move on to the last step, which has the evaluation formula inside. As mentioned before the questions are based on the SUS questionnaire.

5.2 Results Evaluation

In this chapter, the results of the evaluation formula will be presented. Hereby the results of each question will be presented and analyzed. Lastly, the overall outcome of the evaluation will be discussed.

At the beginning of the evaluation formula, demographic data will be retrieved from the participants. Hereby the age, the degree, the course of study, and the current semester for the mentioned course of study and degree. In figure 36 can be seen the cake diagram of the age distribution, which has five participants between the age of 21 to 23, three participants between 24-26 and two participants between 18-20, and zero participants that are older than 26.

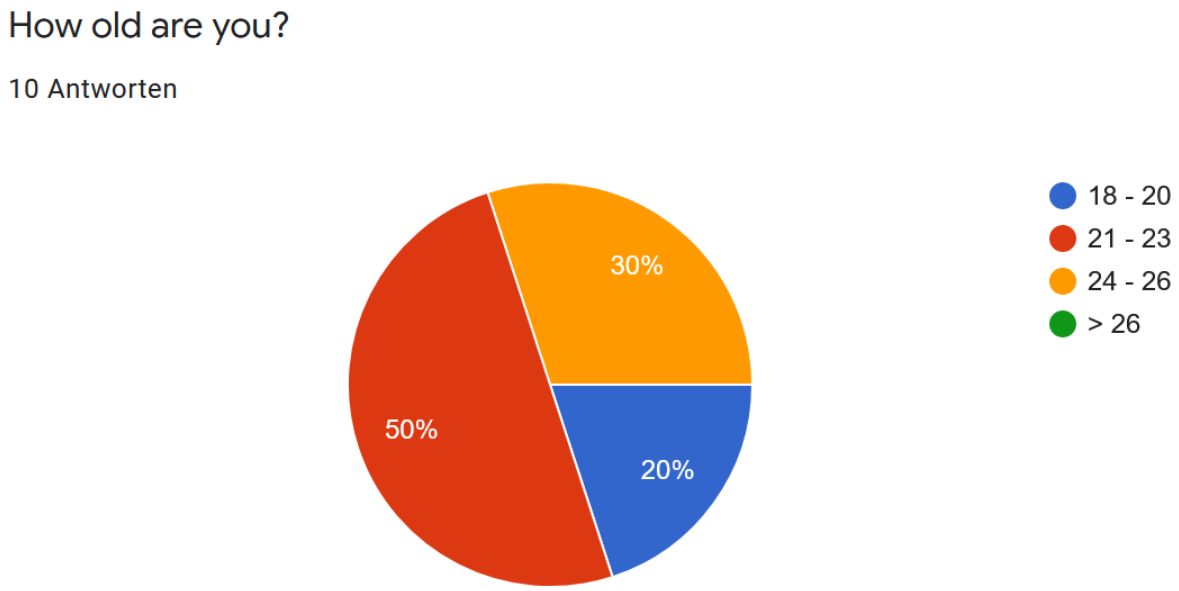


Figure 36: Age Distribution
Source: own creation

Additionally to the age, the demographic data also asks about the current degree that will be achieved within the current course of study. The evaluation results can be seen in Figure 37. Six of the participants are aiming to get a bachelor's degree within their course of study, one participant is aiming for a master's degree, and three participants are aiming for a state examination. It is important to mention that the information on the age distribution correlates with the aimed degree of the study.

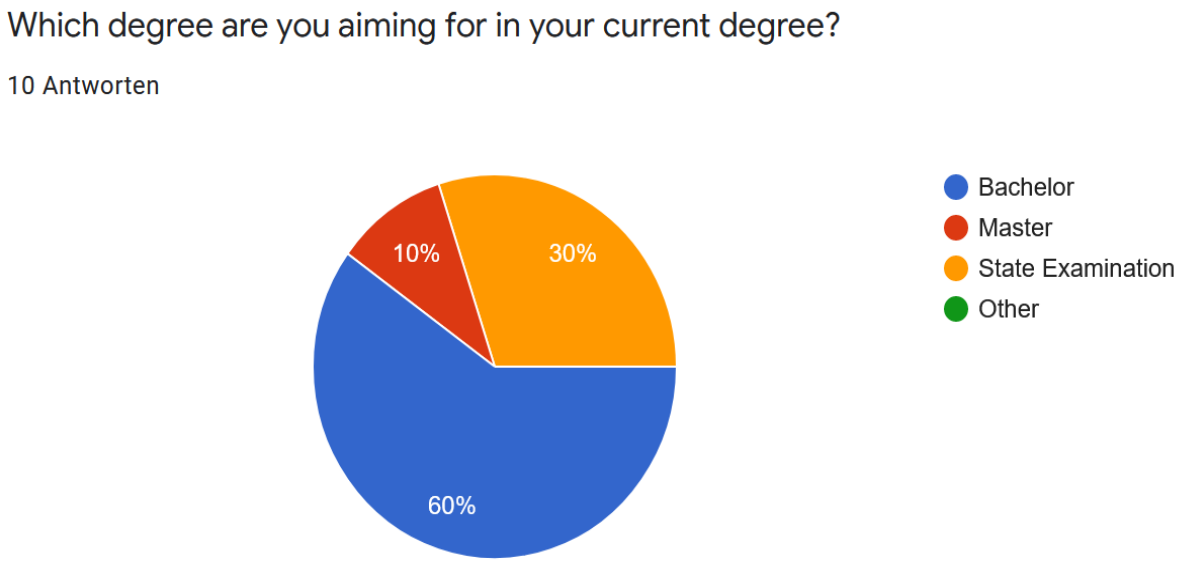


Figure 37: Degree Distribution
Source: own creation

As mentioned before, the current course of study is also asked within the demographic data. Hereby I tried to cover the more important course of studies as click-able items, and an additional item "Other" in order to cover all optionalities. The evaluation shows a 50 / 50 distribution of the more suitable and important course of studies, and the other course of studies. The important course of study is Business Administration, Computer Science, and Information Systems. Figure 38 shows the distribution of the course of study.

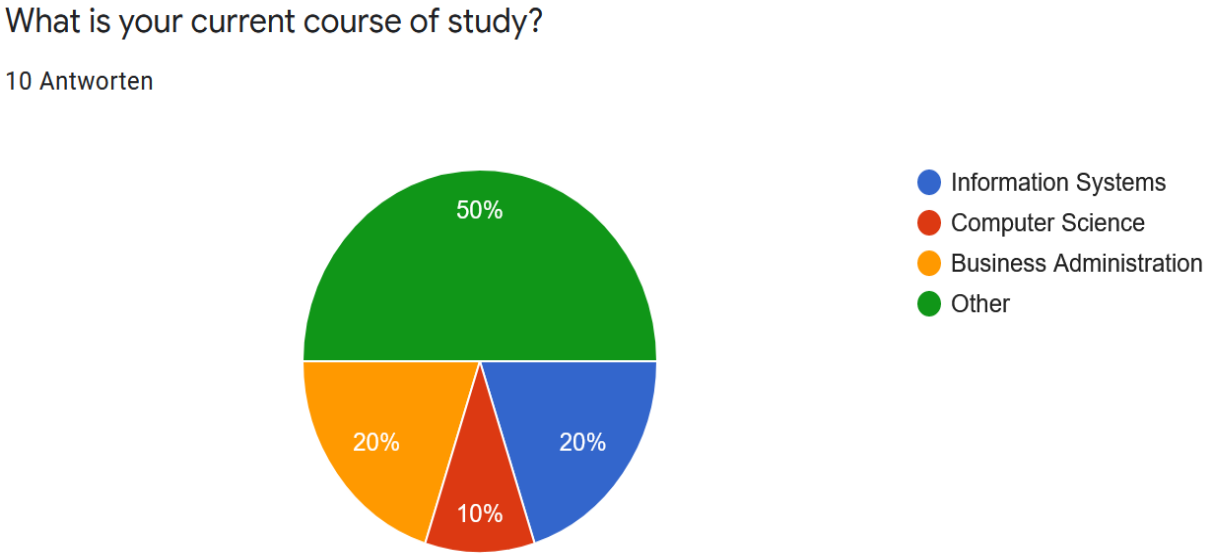


Figure 38: Course of Study Distribution
Source: own creation

The last information about the demographic data will be the current semester within the above-mentioned course of study and degree. Hereby the only answer option was short text optionality. This answer optionality can reproduce a false bar graph when the same answer is typed differently. Nevertheless, every participant give a correct answer, which can be seen in figure 39. Hereby the current semester is distributed almost evenly between the first and ninth semesters.

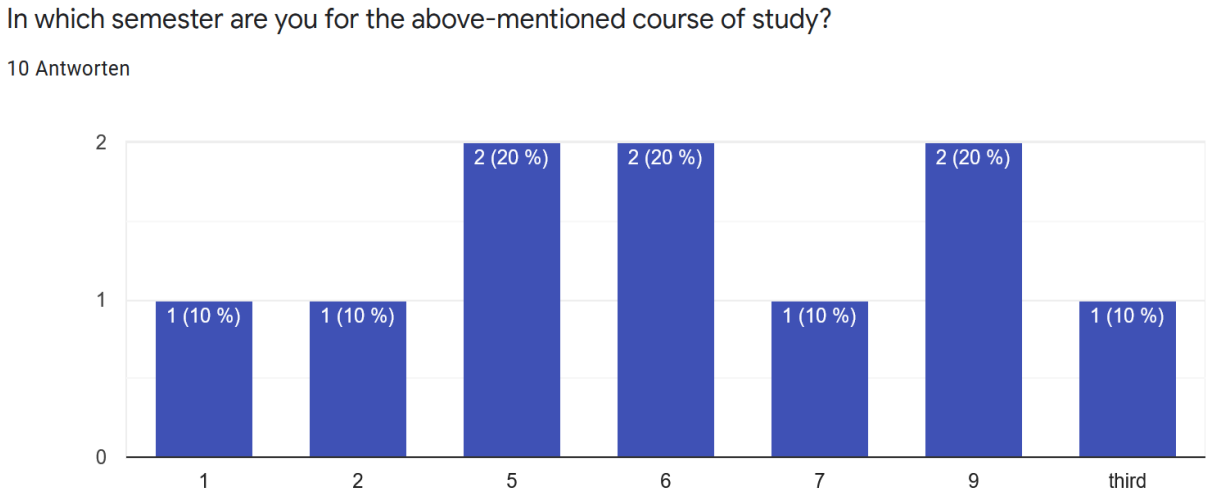


Figure 39: Semester Distribution
Source: own creation

After the demographic data, the SUS questionnaire will be asked. The first out of ten questions was the complexity of the monitoring tool. Hereby the distribution of the answers is mostly positive, which means that the participants selected strongly disagree and disagree. 90% of the students thought that the monitoring tool was not unnecessarily complex. Only one out of 10 participants had a neutral opinion about the complexity. For the calculation of the SUS score the average score for the first question has to be calculated. In this case, the average response has a value of 1,4.

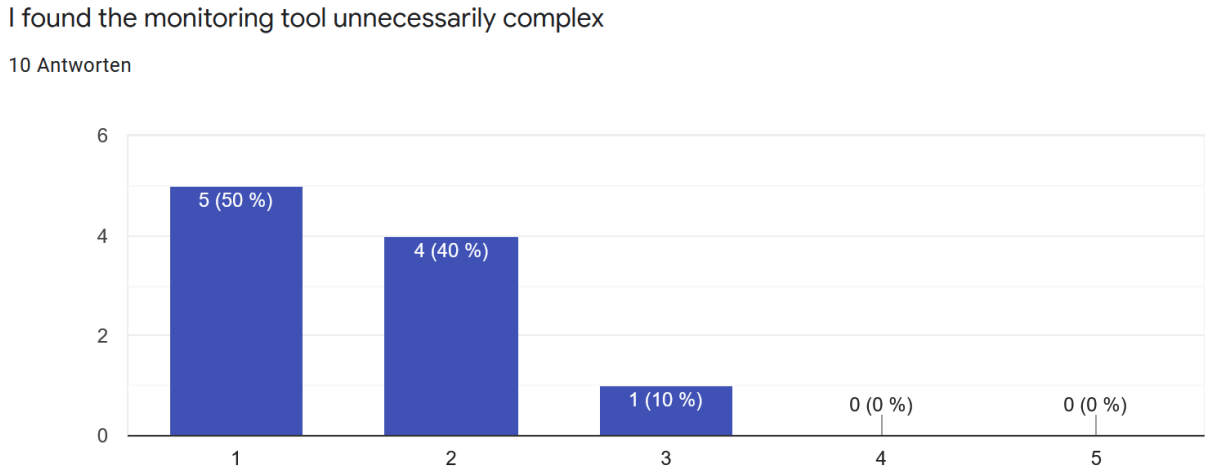


Figure 40: Complexity
Source: own creation

The next question was about the regular usage of the monitoring tool. Hereby the distribution was not as strongly supported as the first question, but here almost 70% had the opinion that they strongly agree and agree on the regular usage of the monitoring tool. It is important to note that only one participant would use the monitoring tool frequently. 30% of the participants have a neutral view of the usage of the monitoring tool. The average response of this question has a value of 3,8.

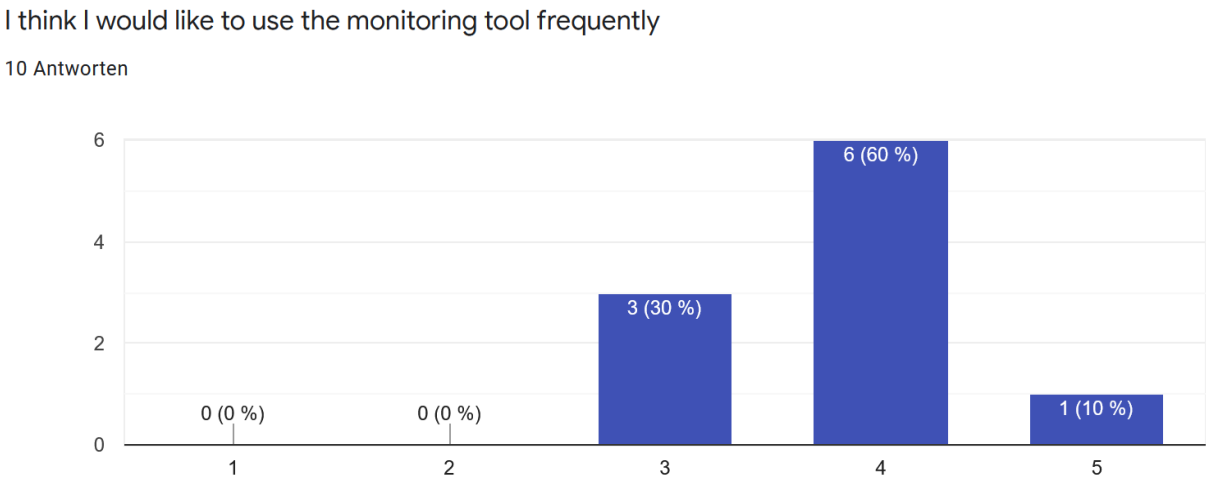


Figure 41: Regular Usage
Source: own creation

The next question was about the easy usage of the monitoring tool. This is moreover a question that also correlates with the complexity question. Hereby we can see that 70% of the students agree with the statement. Hereby one student strongly disagrees with the statement. It is important to note, that this answer can be a fraudulent or accidentally wrong input. However, 30% of the participants did not support the easiness of the monitoring tool. The following figure showcases the distribution within this question. This question has an average score of 4,1.

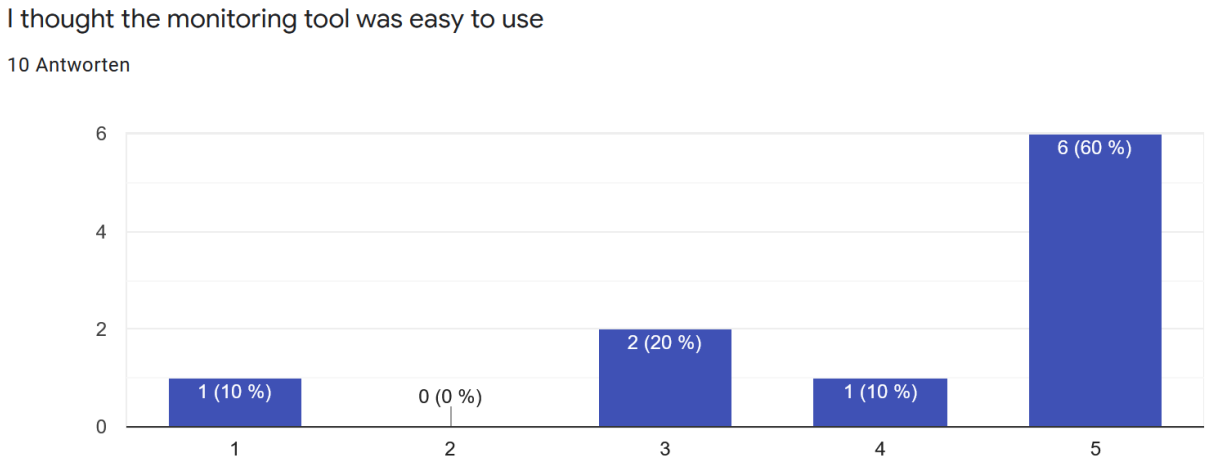


Figure 42: Easy to Use
Source: own creation

The fourth question was about the help of a technical person while using the monitoring tool. Hereby the participants also had positive feedback about the monitoring tool. The distribution was clearly one-sided. 90% of the participants thought that they would not need help to use the monitoring tool. The distribution is visualized in figure 43. This question has an average score of 1,7.

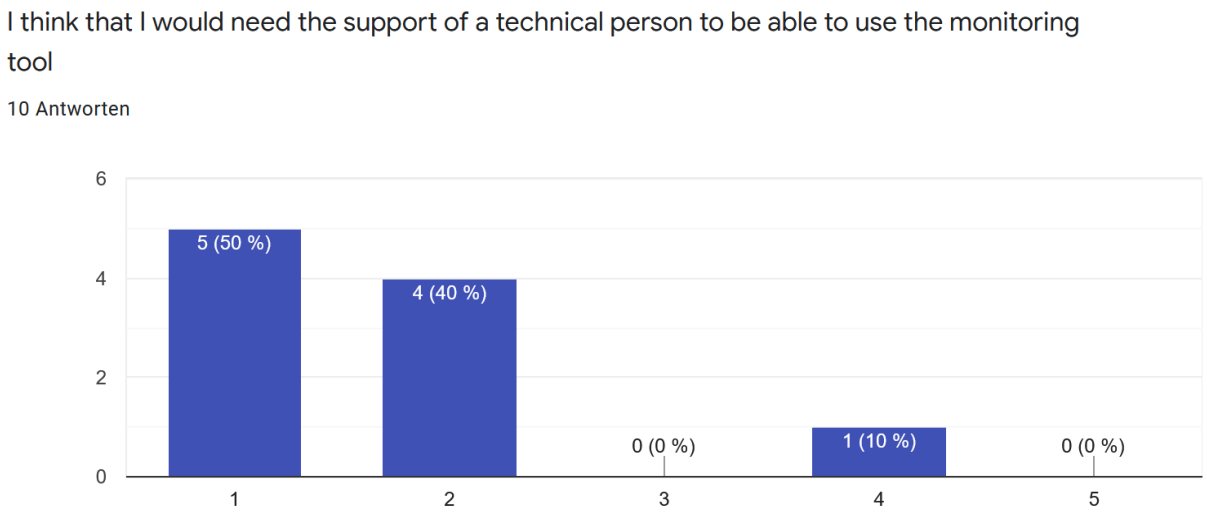


Figure 43: Help for Usage
Source: own creation

The next question was about the integration of the functionalities within the monitoring tool. Hereby a clear positive distribution can be seen. None of the participants disagreed with this statement, and 90% agreed to the statement. Only one participant had a neutral perspective on the integration of the functionalities in the monitoring tool. Hereby the average score of this question is 4,2.

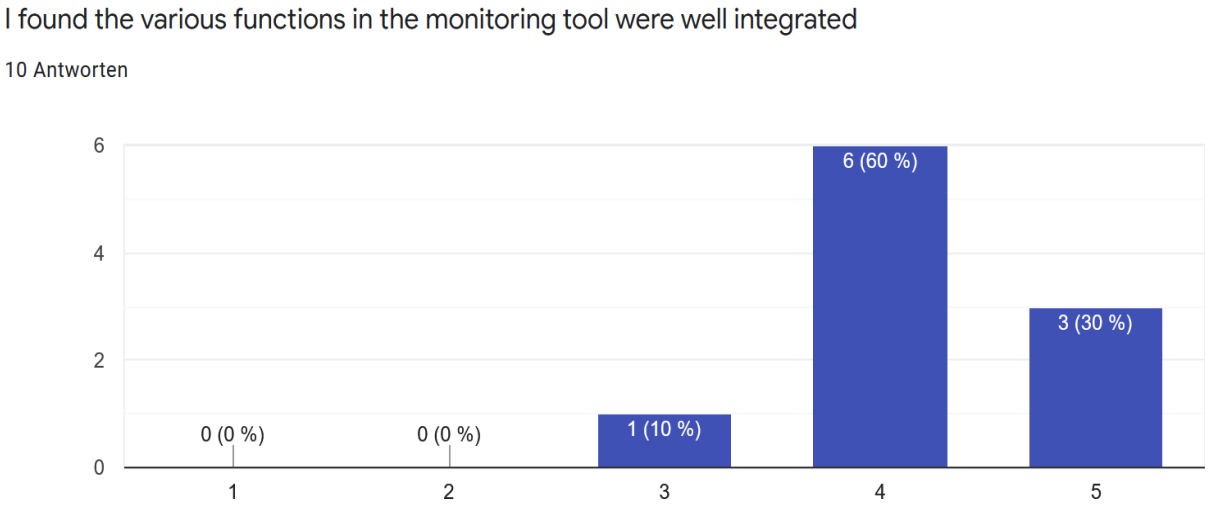


Figure 44: Integration of the functionalities
Source: own creation

Furthermore, the sixth question was about the inconsistency within the monitoring tool. Hereby the results have also a very clear distribution. 80 % disagree with this statement, which is a very positive result for the monitoring tool, and one participant agrees with the statement. The response value of this question has a score of 1,6.

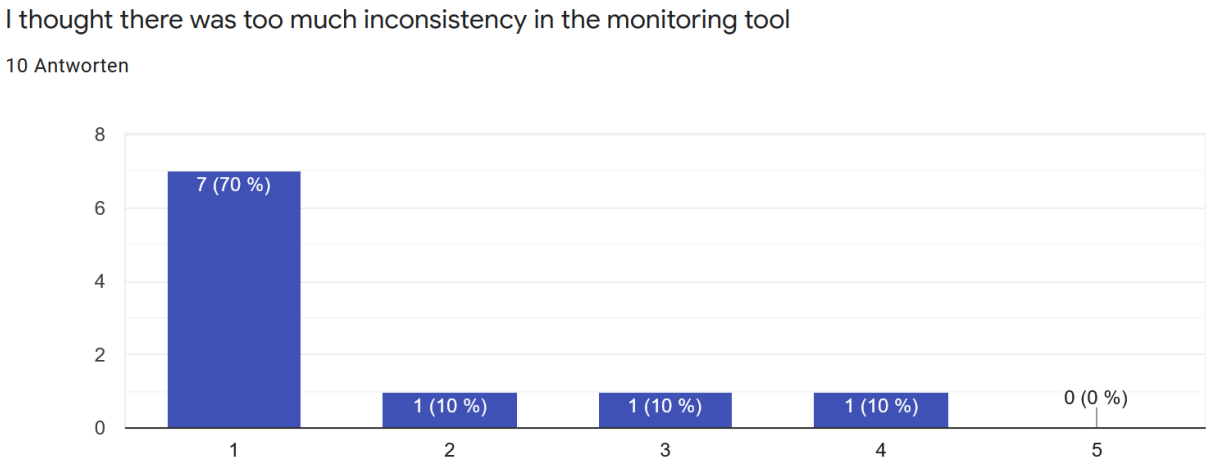


Figure 45: Inconsistency of the monitoring tool
Source: own creation

The next question was about the overall learning time to use the monitoring tool. Here the results are more distributed. 20% disagree and 70% agree with the statement. The average response score of this question has a value of 4,0.

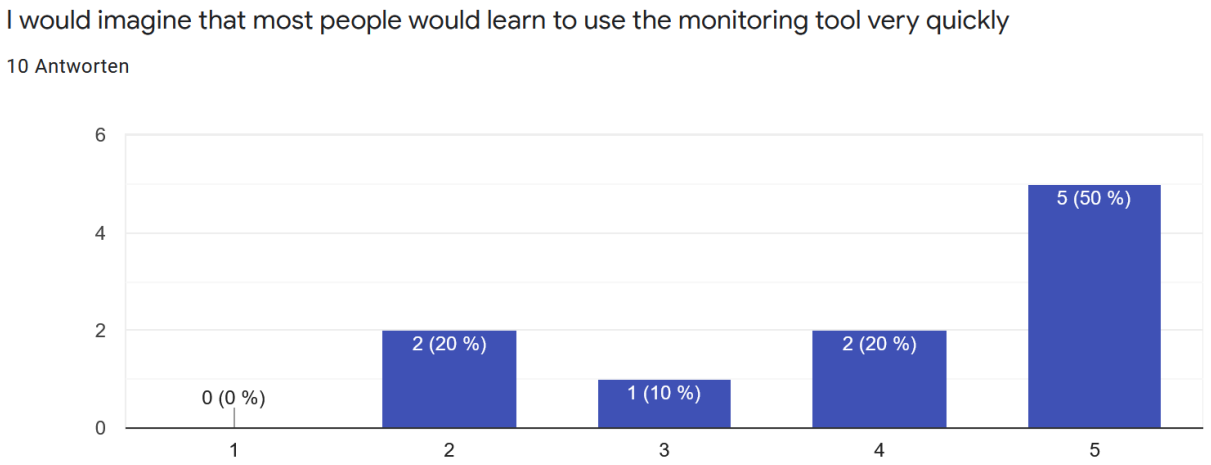


Figure 46: Quick learning
Source: own creation

The next question is about the cumbersome usage of the monitoring tool. Here the distribution was again very one-sided. Only 20% of the Participants find the monitoring tool cumbersome to use. The average response value within this question is 2,1.

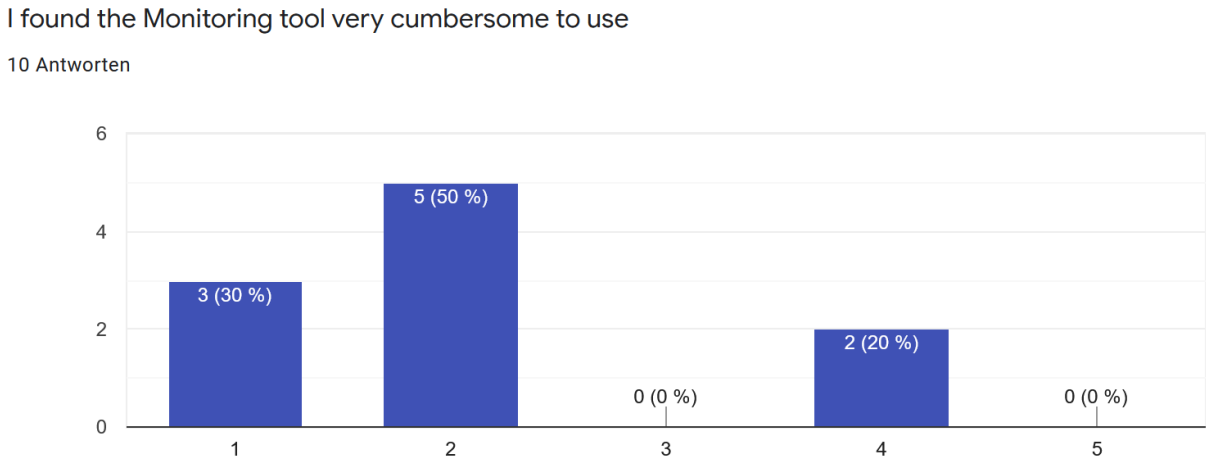


Figure 47: Cumbersome Usage
Source: own creation

Furthermore, the next question was about confidence while using the monitoring tool. Hereby the majority of the participants agree with this statement. 20% of the participants had a neutral perspective and only one participant disagreed with this statement. The corresponding graph is visualized in figure 48. The average response value within this question is 3,9.

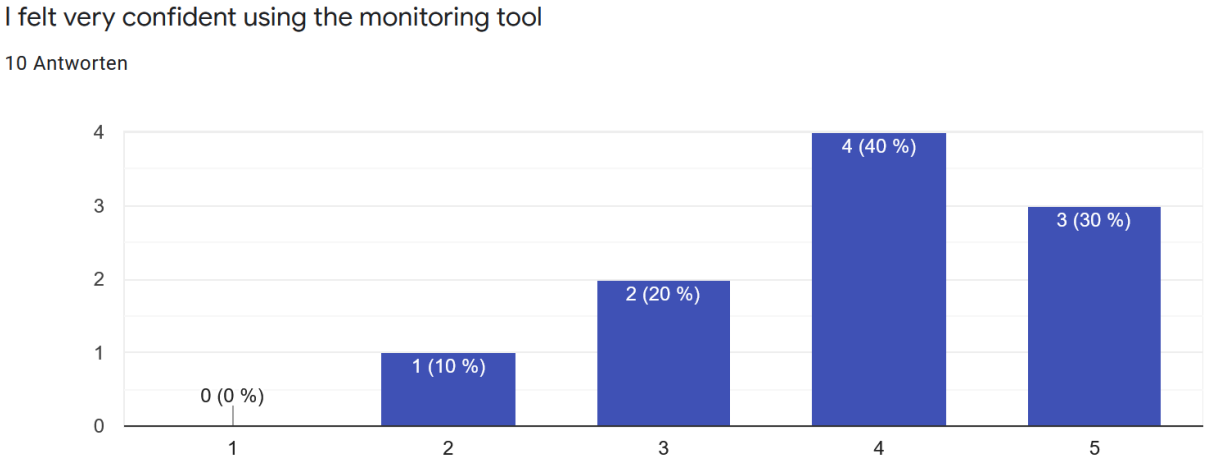


Figure 48: Confidence while using the monitoring tool
Source: own creation

The last question of the SUS evaluation questionnaire was about the preknowledge before using the monitoring tool. Hereby the distribution had a peek at the disagree answer optionality. Hereby the average response value is 2,1.

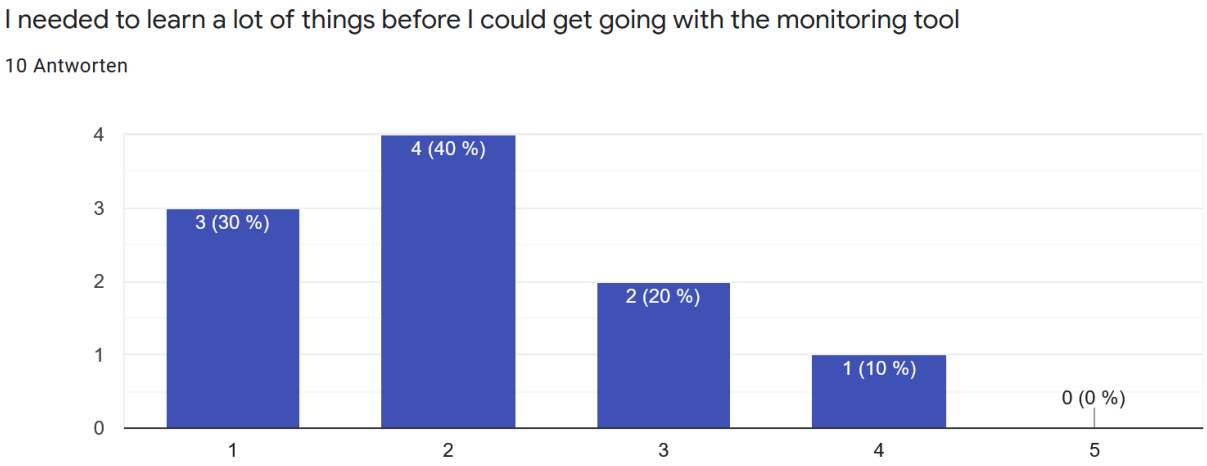


Figure 49: Preknowledge before using the monitoring tool
Source: own creation

The calculation of the SUS score is visualized in the following table. Hereby the averages score of each question has been identified and calculated. On the left-hand side of the table, the odd questions are gathered, and on the right-hand side, the even-numbered questions are gathered. The reason for that is because the odd-numbered questions have different positive responses, which is strongly agree, whereas the even-numbered questions have the positive feedback strongly disagree. After that, the sum of the categorized questions needed to be calculated. The total sum of odd-numbered questions gets subtracted by 5 which will give the corresponding result value. For the even-numbered questions, the value 25 needs to be subtracted by the total sum of even-numbered questions. Lastly, both values need to be added and then multiplied by 2,5. The result of this will give the SUS score, which is 77,75 for the monitoring tool. The maximum value of the SUS score would be 100. Figure 50 showcases the SUS score grading scale. The figure shows the marginal value between a SUS score of 50 and 68, where the acceptable range start also at 68. The value of the monitoring tool is almost 80, which means that the SUS score of the monitoring tool falls between good and excellent (Bangor, Kortum, & Miller, 2009).

Question 1	3,8	Question 2	1,4
Question 3	4,1	Question 4	1,7
Question 5	4,2	Question 6	1,6
Question 7	4,0	Question 8	2,1
Question 9	3,9	Question 10	2,1
Total Odd:	20	Total Even:	8,9
Result:	20 – 5 = 15	Result:	25 – 8,9 = 16,1
SUS Score	(15 + 16,1) * 2,5 = **77,75**		

Table 11: SUS Score calculation
Source: own analysis

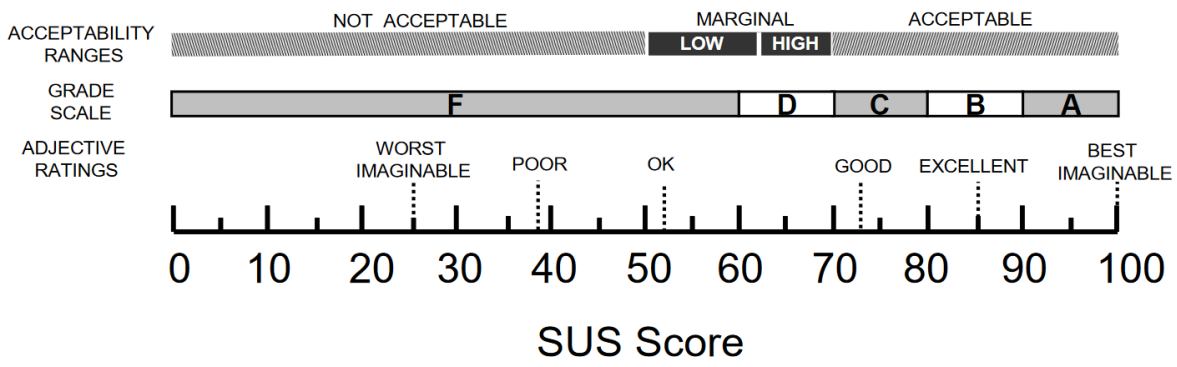


Figure 50: SUS Score Interpretation
Source: (Bangor et al., 2009)

6 Conclusion and Outlook

In this section, I will discuss the overall progress and results of the research questions that have been realized inside this thesis. Afterwards, I will talk about the future work and additional functionalities and improvements, that can be undertaken within the implementation of the monitoring tool.

6.1 Results and Discussion

As stated in the beginning, 3 different research questions have been created and researched within this thesis. These are:

- **Research Question 1:** *What are the technical requirements for a Monitoring Tool in an e-learning environment?*

- **Research Question 2:** How can the elicited requirements for a Monitoring Tool in an e-learning environment be implemented?

- **Research Question 3:** How much does the developed Monitoring Tool fulfill the technical requirements?

In the first research question, I have conducted a literature review based on the methodology of Brocke et al. (2009) about the technical requirements of a student performance monitoring tool. The results of the literature review have been combined with analyses of existing solutions. After gathering all requirements, the requirements have been tailored for the needs of the SAP UCC. Most of the authors and researchers within the literature review focused on requirements that are more general-based requirements, and not categorized into technical requirements. The challenge hereby was to elicit the technical requirements from the literature.

After the requirements elicitation, I have implemented the monitoring tool within the case studies, that were assigned to me, whereas Henryk Mustroph implemented his assigned case studies. For the implementation methodology, we decided to use Scrum with an additional Kanban board. Therefore we had sprint planning meetings and sprint retro perspective meetings, which added value for the implementation and review of the monitoring tool. It also increased the interaction and communication within the implementation process. It is important to note that Mr. Mustroph is hereby working on his thesis from another perspective on the monitoring tool. The technology that we used was the SAPUI5 framework for the frontend, where we implemented XML and javascript code on the SapWebIDE. For the backend, we used the ABAP language within the SAP GUI (LOGON). The two components are communicated via the REST interface. The monitoring tool has been implemented for two different views. On the one hand, the lecturer’s perspective, which has its own UI and other restrictions, and on the other hand, the student’s view.

After the implementation of the monitoring tool, the evaluation process has been conducted. Hereby it is important to mention that Mr. Mustroph did his own evaluation based on the lecturer's perspective and didactical aspects of the monitoring tool, whereas I did the evaluation based on the technical aspects and usability of the monitoring tool. Therefore I created a SUS evaluation methodology, which basically is a predefined evaluation formula with questions based on the user experience on the evaluated system. The participants in the evaluation process of the monitoring tool were not restricted on their course of study, the only restriction was that

the participants are currently students in universities. Before the participants examine the monitoring tool and answer the evaluation formula, I created a presentation within a video, where I explain the overall motivation and methodology of the evaluation process. Thereby information like the SAP UCC, the case studies, and the digital transformation process were given. The outcome result of the evaluation was 77,75, which is a value between good and excellent. The conclusion hereby is that the monitoring tool has good usability, but still can be improved.

6.2 Future Work

In this chapter, I will talk about the possible changes and future works on the monitoring tool. First, it is important to mention that the SAP UCC has a total of 5 case studies. Within this thesis and Mr. Mustrophs thesis the case studies Materials Management, Production Planning, Enterprise Asset Management, Finance & Controlling got implemented and integrated into the monitoring tool. This means that there exists one case study from the SAP UCC, which is not integrated into the monitoring tool. Hereby the case study is Sales & Distribution. This can still be integrated into the monitoring tool without further changes. Another aspect of improvement is within the comment functionalities. Hereby the monitoring tool can be improved with push notifications, which means that the comments and messages should be sent in real-time, and give a small notification to the corresponding user if a new comment gets send to him/her.

7 Bibliography

Adams, N. E. (2015). Bloom's taxonomy of cognitive learning objectives. Journal of the Medical Library Association: JMLA, 103(3), 152.

Alowayr, A., & Badii, A. (2014). Review of monitoring tools for e-learning platforms. arXiv preprint arXiv:1407.2437.

Baumgartner, P., Häfele, H., & Maier-Häfele, K. (2002). E-Learning Standards aus didaktischer Perspektive: na.

Brocke, J. v., Simons, A., Niehaves, B., Niehaves, B., Reimer, K., Plattfaut, R., & Cleven, A. (2009). Reconstructing the giant: On the importance of rigour in documenting the literature search process.

Glossary, E. (2015). Curriculum. Retrieved from https://www.edglossary.org/curriculum/

Graf, S. (2007). Adaptivity in learning management systems focussing on learning styles.

Lithuania, N. A. O. o. (2016). Monitoring of Education. In.

Mazza, R., & Milani, C. (2004). Gismo: a graphical interactive student monitoring tool for course management systems. Paper presented at the International Conference on Technology Enhanced Learning, Milan.

Müller-Bloch, C., & Kranz, J. (2015). A framework for rigorously identifying research gaps in qualitative literature reviews.

Radha, R., Mahalakshmi, K., Kumar, V. S., & Saravanakumar, A. (2020). E-Learning during lockdown of Covid-19 pandemic: A global perspective. International journal of control and automation, 13(4), 1088-1099.

Rupp, C., Simon, M., & Hocker, F. (2009). Requirements engineering und management. HMD Praxis der Wirtschaftsinformatik, 46(3), 94-103.

Teich, I., Kolbenschlag, W., & Reiners, W. (2008). Der richtige Weg zur Softwareauswahl: Lastenheft, Pflichtenheft, Compliance, Erfolgskontrolle: Springer-Verlag.

Wagner, S., Fernández, D. M., Felderer, M., & Kalinowski, M. (2017). Requirements engineering practice and problems in agile projects: results from an international survey. arXiv preprint arXiv:1703.08360.

Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of usability studies*, *4*(3), 114-123.

Bass, R. B., Pejcinovic, B., & Grant, J. (2016). *Applying Scrum project management in ECE curriculum.* Paper presented at the 2016 IEEE Frontiers in Education Conference (FIE).

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., . . . Jeffries, R. (2001). Manifesto for agile software development.

Brocke, J. v., Simons, A., Niehaves, B., Niehaves, B., Reimer, K., Plattfaut, R., & Cleven, A. (2009). Reconstructing the giant: On the importance of rigour in documenting the literature search process.

Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry, 189*(194), 4-7.

Cohen, D., Lindvall, M., & Costa, P. (2003). Agile software development. *DACS SOAR Report, 11*, 2003.

Daif, A. R., & Rizkaa, M. A. (2013). *An enhanced model for monitoring learners' performance in a collaborative e-Learning environment.* Paper presented at the 2013 second International Conference on e-Learning and e-Technologies in Education (ICEEE).

Gerard, J., & Katz, S. (2017). SAP S/4 HANA–Implementierungsszenarien. In *In-Memory-Datenbank SAP HANA* (pp. 61-105): Springer.

Kaya, A., Ozturk, R., & Gumussoy, C. A. (2019). Usability measurement of mobile applications with system usability scale (SUS). In *Industrial engineering in the big data era* (pp. 389-400): Springer.

Meyen, E. L., Aust, R. J., Bui, Y. N., & Isaacson, R. (2002). Assessing and monitoring student progress in an e-learning personnel preparation environment. *Teacher education and special education, 25*(2), 187-198.

Mircea, E. (2019). Project management using Agile frameworks. *Academy of Economic Studies. Economy Informatics, 19*(1), 34-44.

quanto-solutions (Producer). (2021). SAP Fiori und Micro Frontends. Retrieved from https://quanto-solutions.de/sap-fiori-und-micro-frontends/

Röder, H. (2012). *Specifying usability features with patterns and templates.* Paper presented at the 2012 First International Workshop on Usability and Accessibility Focused Requirements Engineering (UsARE).

Software, M. G. (Producer). (2021). Mountain Goat Software. *Mountain Goat Software*. Retrieved from https://www.mountaingoatsoftware.com/agile/scrum/resources/overview

UCC, S. (Producer). The Digital Transformation of Global Bike. *The Digital Transformation of Global Bike*. Retrieved from http://i04lp1.informatik.tu-muenchen.de/dtcurriculum/index.html?sap-client=101&sap-language=DE#/home

Zhou, W., Li, L., Luo, M., & Chou, W. (2014). *REST API design patterns for SDN northbound API.* Paper presented at the 2014 28th international conference on advanced information networking and applications workshops.

Appendix

Monitoring Tool Evaluation

This evaluation has the goal to get information about improvements of the monitoring tool, that was implemented as part of our Bachelors Thesis. Additionally, the user experience will be analyzed. The Evaluation is based on the System Usability Scale Evaluation methodology.

* Erforderlich

General Information

1. How old are you?

Markieren Sie nur ein Oval.

- 18 - 20
- 21 - 23
- 24 - 26
- > 26

2. Which degree are you aiming for in your current degree?

Markieren Sie nur ein Oval.

- Bachelor
- Master
- State Examination
- Other

3. What is your current course of study?

Markieren Sie nur ein Oval.

- ◯ Information Systems
- ◯ Computer Science
- ◯ BusinessAdministration
- ◯ Other

4. In which semester are you for the above-mentioned course of study?

System Usability Scale

5. I found the monitoring tool unnecessarily complex *

Markieren Sie nur ein Oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

6. I think I would like to use the monitoring tool frequently *

Markieren Sie nur ein Oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

7. I thought the monitoring tool was easy to use *

Markieren Sie nur ein Oval.

	1	2	3	4	5	
Strongly disagree	◯	◯	◯	◯	◯	Strongly agree

8. I think that I would need the support of a technical person to be able to use the monitoring tool *

Markieren Sie nur ein Oval.

	1	2	3	4	5	
Strongly disagree						Strongly agree

9. I found the various functions in the monitoring tool were well integrated *

Markieren Sie nur ein Oval.

	1	2	3	4	5	
Strongly disagree						Strongly agree

10. I thought there was too much inconsistency in the monitoring tool *

Markieren Sie nur ein Oval.

	1	2	3	4	5	
Strongly disagree						Strongly agree

11. I would imagine that most people would learn to use the monitoring tool very quickly *

Markieren Sie nur ein Oval.

	1	2	3	4	5	
Strongly disagree						Strongly agree

12. I found the Monitoring tool very cumbersome to use *

Markieren Sie nur ein Oval.

	1	2	3	4	5	
Strongly disagree						Strongly agree

13. I felt very confident using the monitoring tool *

Markieren Sie nur ein Oval.

	1	2	3	4	5	
Strongly disagree						Strongly agree

14. I needed to learn a lot of things before I could get going with the monitoring tool *

Markieren Sie nur ein Oval.

	1	2	3	4	5	
Strongly disagree						Strongly agree