DHBW 2025

Software Requirements Specification

for

RandExGen

Version 0.1 approved

Prepared by Niklas Reusch

Created on 07.10.2025

Table of Contents

Revision History

Name	Date	Release Description	Version
Niklas Reusch	07.10.2025	Initial version of the *RandExGen* Software Requirements Specification (SRS) for the DHBW Stuttgart project in Software Engineering 1.	0.1

1. Introduction

1.1 Purpose

This Software Requirement Specification (SRS) defines the functional and non-functional requirements for the software product RandExGen – Version 0.1. The RandExGen Software is a JavaFX-implemented desktop application developed as part of the Software Engineering 1 module at DHBW Stuttgart. The name RandExGen is a composition out of three words: Random, exam and generator, which show the purpose of this application: The generation of randomized exams and practice exams from a user-uploaded file containing tasks and subtasks.

1.2 Document Conventions

This SRS is entirely written in the English language. To ensure that all requirements are prioritized correctly and written precisely, the documentation follows some formatting and writing rules:

Font and Style

- Headings are written in Liberation Sans font and have a font size of 18
- Subheadings are written in Liberation Sans font and have a font size of 14
- Subheadings inside of text are written in Carlito font, are underlined and have a font size of 11
- Text is written in Carlito font and have a font size of 11

Requirement Identification

- Every functional requirement is identified with FREQ-X, where X is the number of the requirement
- Every non-functional requirement is identified with NFREQ-X, where X is the number of the requirement

Priority levels

- Each requirement has its own priority. Some of them are optional and many are necessary.
- May – Optional
- Must – Necessary

Text, Diagrams and Pictures

- Use Cases are shown in form of Use-Case-Diagrams
- Java Code is shown in Class-Diagrams
- UI Structure is shown in form of pictures
- Technical terms are shown in form of tables
- The rest is written in text

1.3 Intended Audience and Reading Suggestions

The primary audiences of the RandExGen Software are:

Developers: Responsible for implementing the software according to the requirements described in this SRS. They will mainly focus on the Functional Requirements, System Requirements, and Design Constraints sections.

Testers: Use the requirements defined in this document to verify that all functionalities of the software are correctly implemented. Their main focus is on the Functional and Non-functional Requirements as well as the System Scenarios.

Future Contributors and Documentation Writers: May use this SRS as a reference for updating the software to future versions. They will primarily read the Overview, Design Constraints, and Appendices.

Users: Represent lecturers or exam creators who use the application to generate exams and practice exams. They are primarily interested in the user interface, functionality, and usability aspects described in the Product Functions and User Interface sections.

1.4 Product Scope

- The software must be implemented with JavaFX
- The English language must be the default language of the software
- The application must be operated through a graphical user interface (GUI)
- The system must provide randomized generation of exams and practice exams
- The user must have the option to load, modify and save .XML-Files
- The solution of the generated exam must also be generated
- The generated files must be .PDF-Files
- The application shall be open source and executable locally without internet connection or other dependencies
- The application is created for desktops and not for mobile operation systems

2. Overall Description

2.1 Product Perspective

The RandExGen Software has the goal to automate the generation of balanced and randomized exams, while the user still has full control over the exam data. The user has the ability to add, delete, as well as edit exam data and save them as an XML File. Out of a newly created or loaded XML File, a randomized exam or practice exam can be generated. The exam and its solution can be saved on the system as separate PDF Files.

2.2 Product Functions

The RandExGen Software must perform two major options – The first function is the loading, creation and editing of XML Files containing chapters which consist of different subtasks to generate a randomized exam. The second function is the export of the exam and its solution as a PDF File. Every chapter includes at least one subtask from every difficulty level (easy, medium hard). In Addition, a chapter must contain 1/3 subtasks from every difficulty level to make sure that the generated exam is balanced and fair. The user can select an exam type for every subtask (regular exam or practice exam) which will lead to the system including or excluding the subtasks to the generation process. Before exporting the exam as a PDF File, the user can see a preview of the exam to make sure that it hits the expectations.

2.3 Operating Environment

- The RandExGen software is a cross-platform JavaFX desktop application designed to run on all major operating systems.
- It is compatible with Windows (10 or higher), macOS, and Linux distributions that support Java.
- As the application is executed as a .jar file, it requires a Java Development Kit (JDK 17 or higher) with JavaFX support.
- RandExGen runs fully offline and does not require any internet connection.

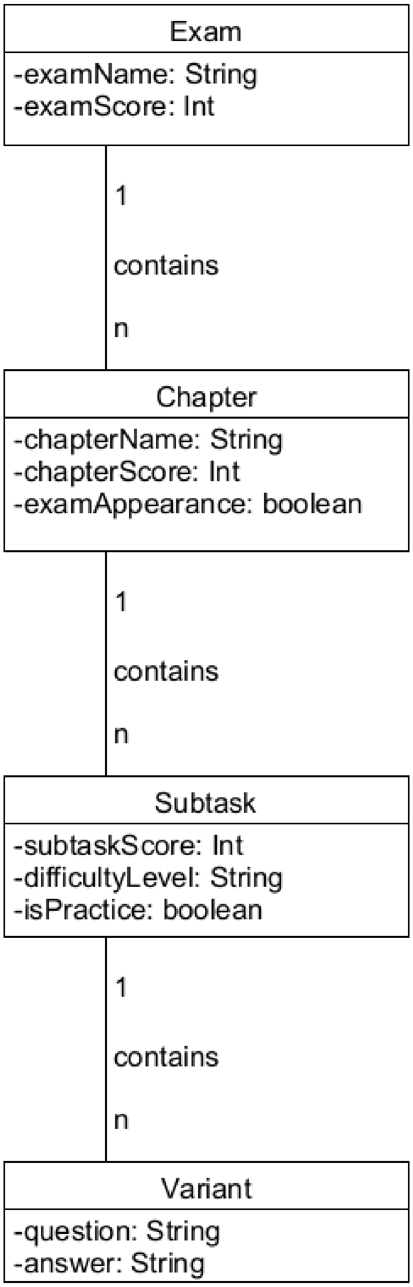
2.4 Design and Implementation Constraints

The RandExGen software must comply with the following technical and design constraints:

- The software must be implemented using JavaFX.
- The default language of the user interface must be English.
- The application must be operated through a graphical user interface (GUI).
- The system must provide a randomized generation of exams and practice exams.
- The user must have the option to load, modify, and save .XML files that contain the task data.
- The solution version of each generated exam must also be created automatically.
- The generated exams and solutions must be exported as .PDF files.
- The software must be open source and executable locally without internet connection or external dependencies.
- The application is designed exclusively for desktop environments and is not intended for mobile operating systems.

2.5 Overview of Exam Data

The diagram below represents the structural data model of the RandExGen software.
An Exam consists of multiple Chapters, each containing several Subtasks, which contain one or several Variants.
Each entity has specific attributes such as the exam name, chapter score, difficulty level, or question and answer of variants.

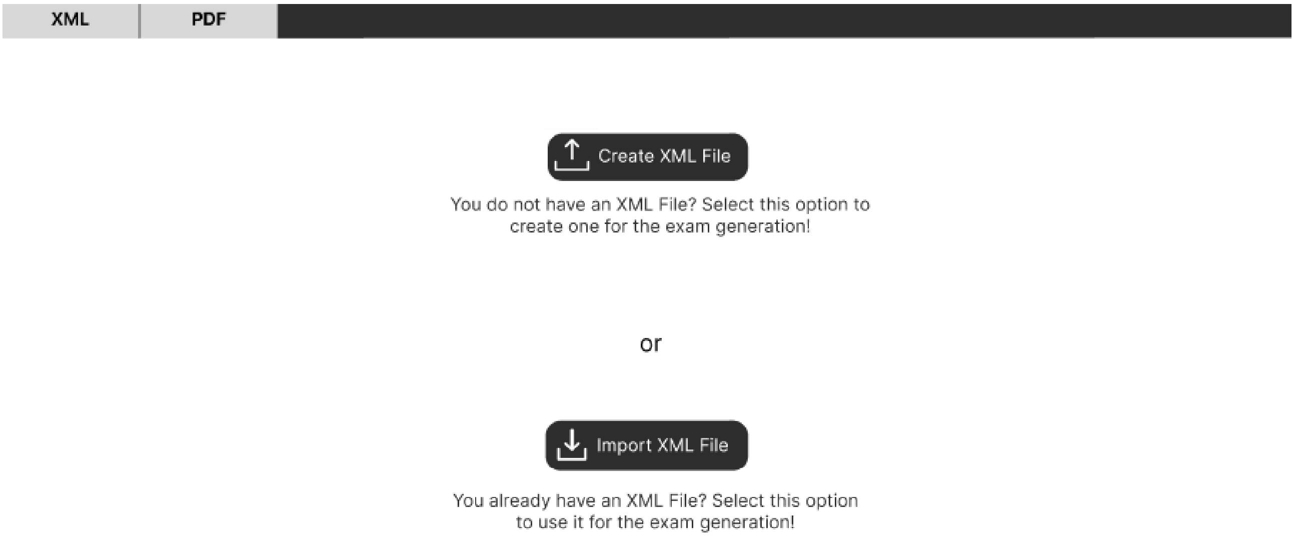


3. External Interfaces and Requirements

3.1 User Interface

The graphical user interface of RandExGen is designed to provide a clear and efficient process for exam generation.
It allows the user to interact with all major system features, which are provided through two different sections of the application – An XML Section and a PDF Section.

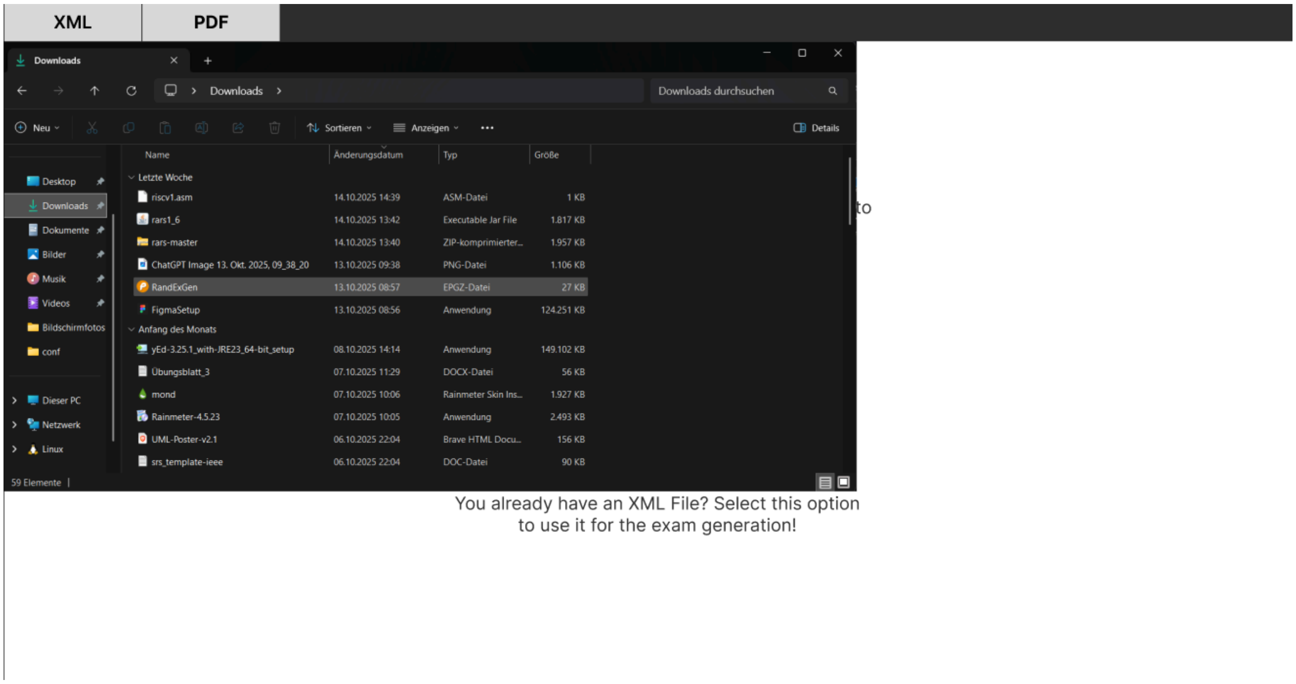
XML Section



You do not have an XML File? Select this option to create one for the exam generation!

You already have an XML File? Select this option to use it for the exam generation!

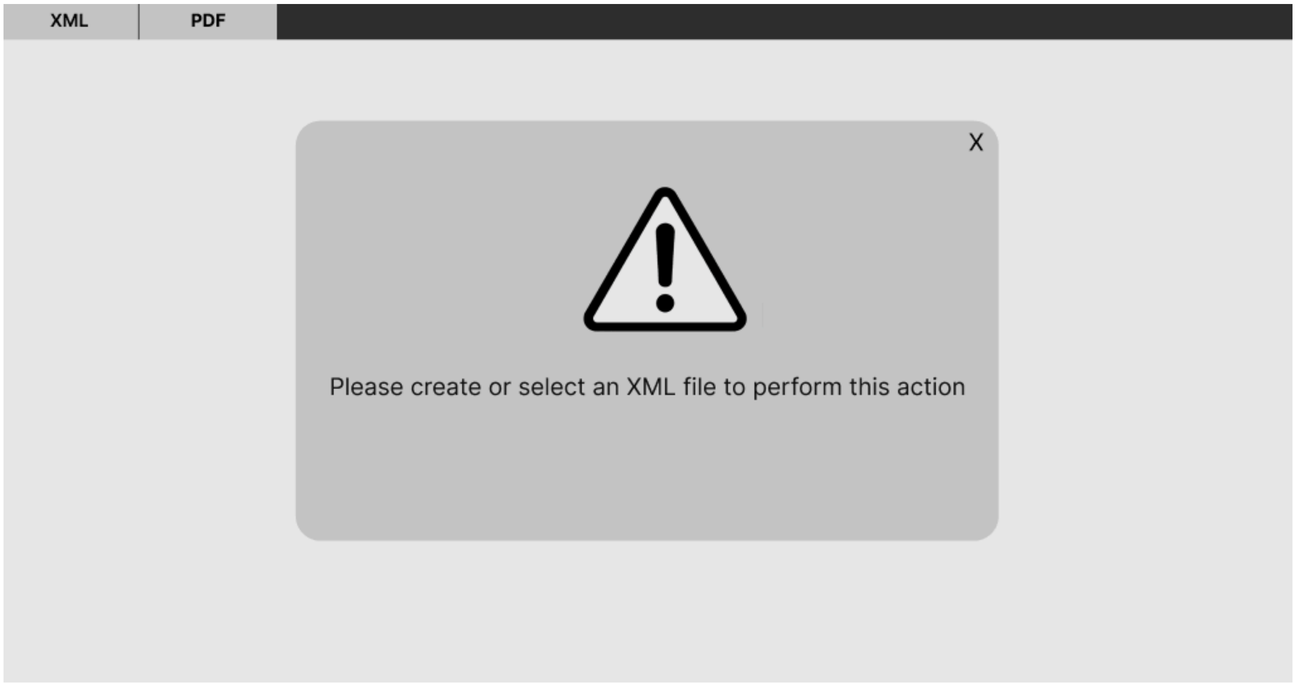
The picture above shows the screen when the user starts the application. If the user already has an XML File on his device, he has the possibility to import it. This happens if the user presses the “Import XML File” button. Otherwise, the user can create an XML File in the application if he presses the “Create XML File” button.



Name	Änderungsdatum	Typ	Größe
Letzte Woche			
riscv1.asm	14.10.2025 14:39	ASM-Datei	1 KB
rars1_6	14.10.2025 13:42	Executable Jar File	1.817 KB
rars-master	14.10.2025 13:40	ZIP-komprimierter...	1.957 KB
ChatGPT Image 13. Okt. 2025, 09_38_20	13.10.2025 09:38	PNG-Datei	1.106 KB
RandExGen	13.10.2025 08:57	EPGZ-Datei	27 KB
FigmaSetup	13.10.2025 08:56	Anwendung	124.251 KB
Anfang des Monats			
yEd-3.25.1_with-JRE23_64-bit_setup	08.10.2025 14:14	Anwendung	149.102 KB
Übungsblatt_3	07.10.2025 11:29	DOCX-Datei	56 KB
mond	07.10.2025 10:06	Rainmeter Skin Ins...	1.927 KB
Rainmeter-4.5.23	07.10.2025 10:05	Anwendung	2.493 KB
UML-Poster-v2.1	06.10.2025 22:04	Brave HTML Docu...	156 KB
srs_template-ieee	06.10.2025 22:04	DOC-Datei	90 KB

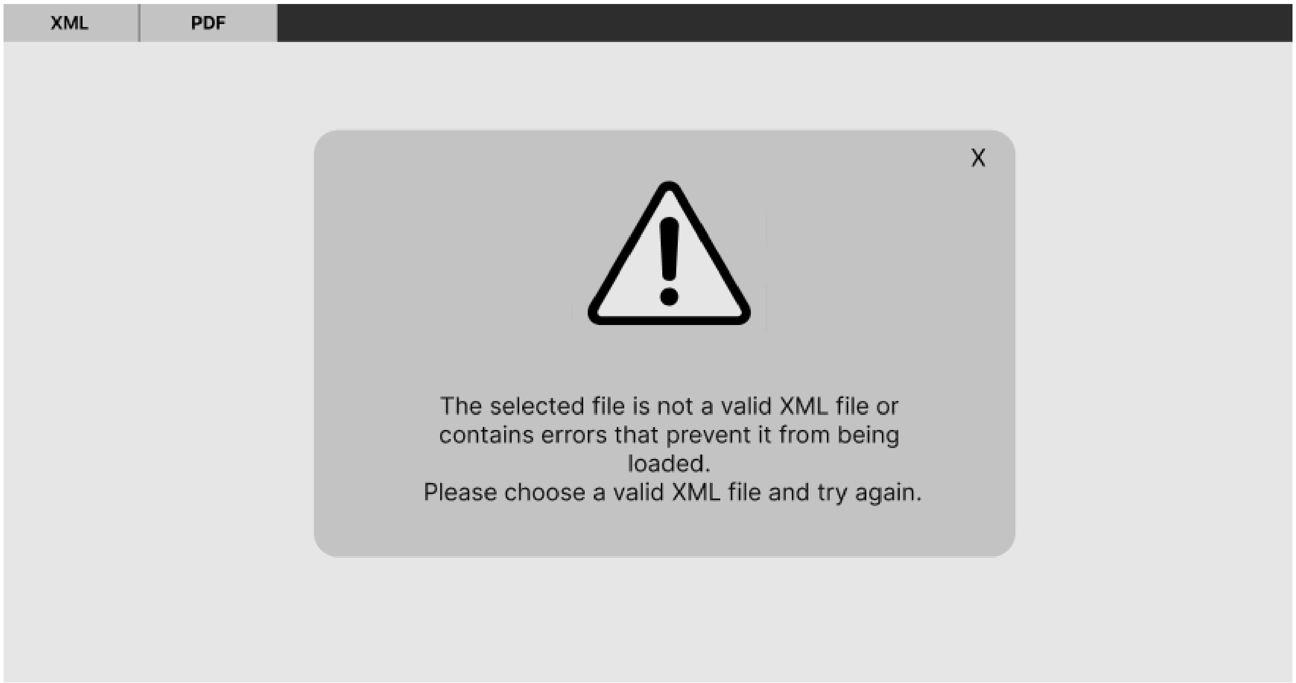
You already have an XML File? Select this option to use it for the exam generation!

The picture above shows the screen after the user presses the “Impot XML File” button. A small explorer window is placed in the top left corner, which gives the user the possibility to navigate through the folders of his system and save the XML File at a specific location.



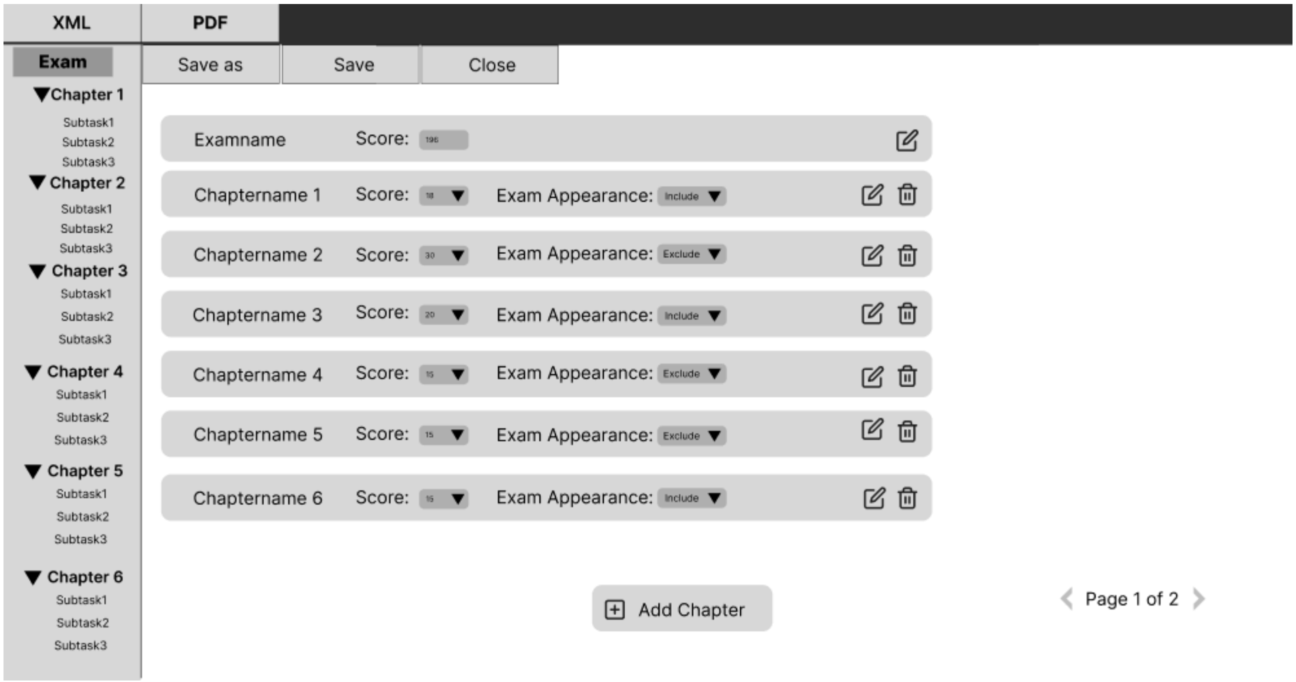
Please create or select an XML file to perform this action

The picture above shows the screen when the user tries to switch to the PDF section through the “PDF” button at the top. The user can see the error message “Please create or select an XML File to perform this action”. It happens, because the PDF section only displays information when the user already uploaded data for the exam generation. As there is no data when there is no XML File imported or created, the application asks the user to do so.



The selected file is not a valid XML file or contains errors that prevent it from being loaded.
Please choose a valid XML file and try again.

The picture above shows the screen when the user loads an XML File which is not of type XML or contains errors. To prevent bugs, the application does not load the selected XML file and tells the user to select a different, valid file. This is shown by the message “The selected file is not a valid XML file or contains errors that prevent it from being loaded. Please choose a valid XML file and try again”.



Examname Score: 196

Chaptername 1 Score: 18 ▼ Exam Appearance: Include ▼

Chaptername 2 Score: 30 ▼ Exam Appearance: Exclude ▼

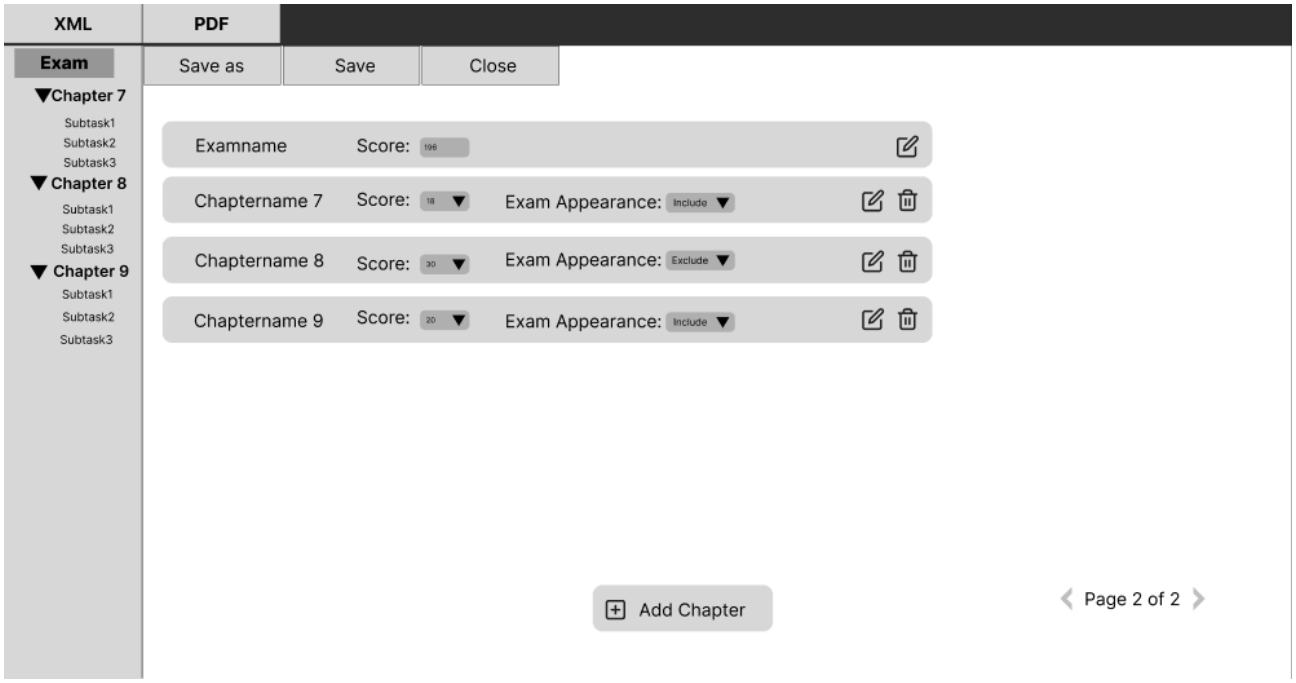
Chaptername 3 Score: 20 ▼ Exam Appearance: Include ▼

Chaptername 4 Score: 15 ▼ Exam Appearance: Exclude ▼

Chaptername 5 Score: 15 ▼ Exam Appearance: Exclude ▼

Chaptername 6 Score: 15 ▼ Exam Appearance: Include ▼

When the user decides to create or import an XML File, he will be in the Chapter Overview Screen, which is shown in the picture above. This screen is split into two halves. The first half is the tree structure on the left side of the screen. It gives the user the possibility to easily navigate through the exam, chapters, and subtasks. The Variants are not displayed in the tree structure, as they have no subordinate exam data. That's why only the Exam, Chapters and the subtasks are shown there. When the user chooses an option, it will be highlighted with a grey background. The second half is the overview on the right side of the screen and displays specific information for what the user chose on the left side. In this screen, the user chose the option "Exam" which will display the title and score of every chapter and the exam. The user has the possibility to edit the score of the chapters, too, but only in a specific range. This happens by clicking on the arrow in the field where the score is shown. The score of the exam cannot be edited, as it is computed by the score of the chapters. Also, the user has the option to change the title of the exam and the chapters, if he presses the edit symbol on the right. In addition, chapters can be completely removed if they shouldn't be part of the randomized exam generation. The user can perform this action by clicking on the trash bin symbol on the right. As the overview is made to create and edit an XML, the user can also add new chapters by clicking on the "Add Chapter" button at the bottom center. In the bottom right corner, the user can switch to the next page to see all chapters. That's why the user will be taken to the last page if he adds a new chapter, because it will be added at the end of the overview. In the overview, there are three more options: Save as, Save and Close. The options "Save as" and "Save" provide that the user can save the created XML File on the system. By clicking the "Close" button, you can go back to the home screen, where you can import or create a new XML File.



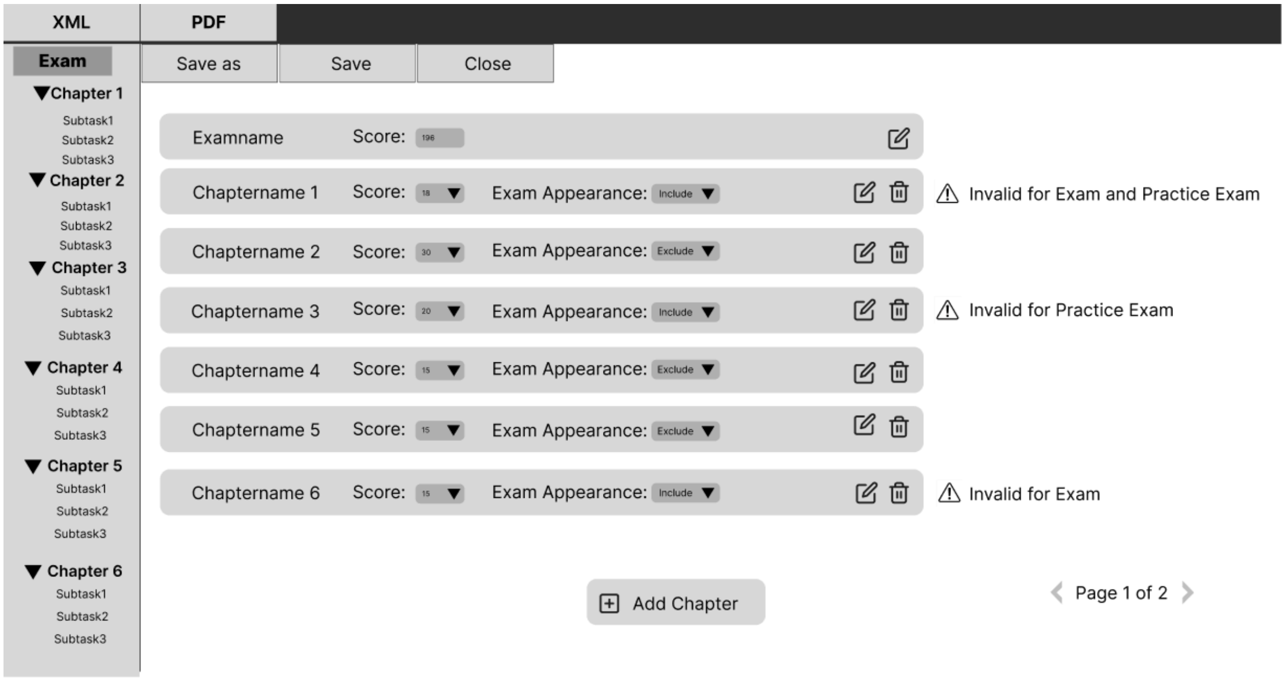
Examname Score: 196

Chaptername 7 Score: 18 ▼ Exam Appearance: Include ▼

Chaptername 8 Score: 30 ▼ Exam Appearance: Exclude ▼

Chaptername 9 Score: 20 ▼ Exam Appearance: Include ▼

The picture above shows what happens when the user switches to the next page. It will change the overview and the tree structure to the remaining chapters. A new page is always created when there is not enough space to display all information in the tree structure and overview.



Examname Score: 196

Chaptername 1 Score: 18 ▼ Exam Appearance: Include ▼ ⚠ Invalid for Exam and Practice Exam

Chaptername 2 Score: 30 ▼ Exam Appearance: Exclude ▼

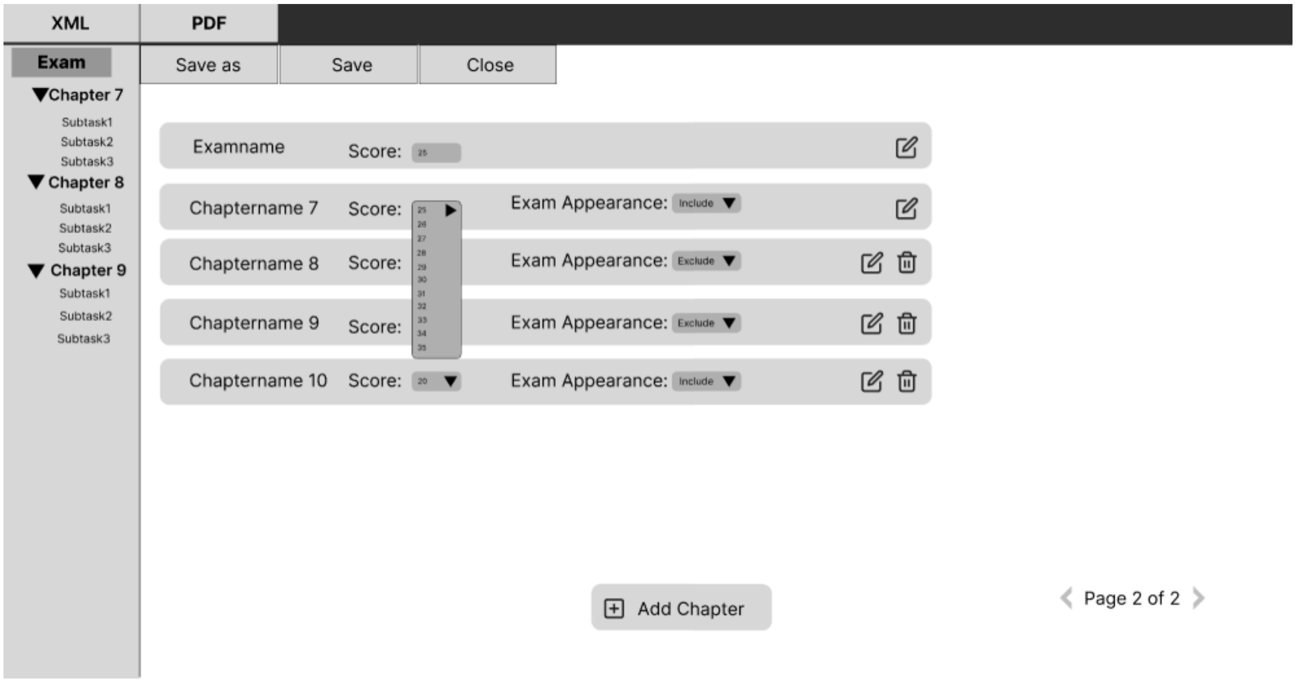
Chaptername 3 Score: 20 ▼ Exam Appearance: Include ▼ ⚠ Invalid for Practice Exam

Chaptername 4 Score: 15 ▼ Exam Appearance: Exclude ▼

Chaptername 5 Score: 15 ▼ Exam Appearance: Exclude ▼

Chaptername 6 Score: 15 ▼ Exam Appearance: Include ▼ ⚠ Invalid for Exam

The picture above shows the error messages, if there are invalid chapters. On the right side of the chapter, the message “Invalid for Exam and Practice Exam” is displayed, if it is invalid for both exam types. “Invalid for Practice Exam” shows that the chapter is invalid to generate a Practice Exam. “Invalid for Exam” is displayed if the chapter is invalid to generate an Exam.



Examname Score: 25

Chaptername 7 Score: 25 ▶ Exam Appearance: Include ▼

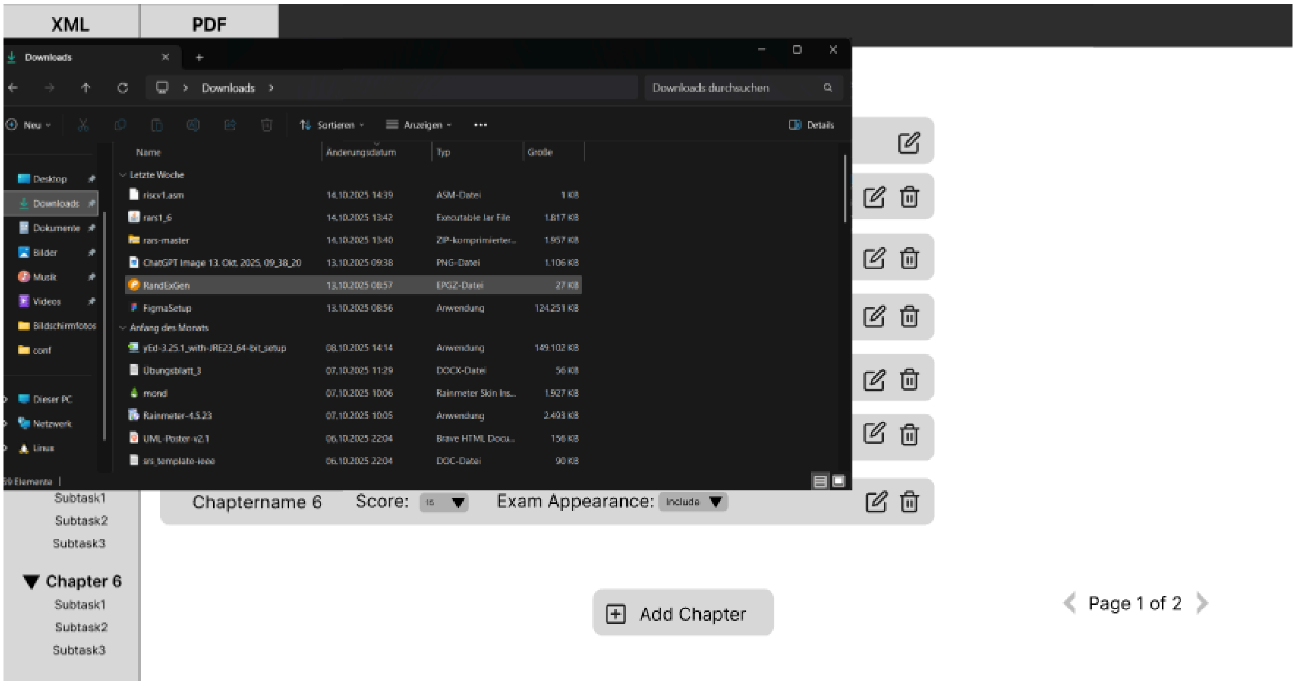
25
26
27
28
29
30
31
32
33
34
35

Chaptername 8 Score: Exam Appearance: Exclude ▼

Chaptername 9 Score: Exam Appearance: Exclude ▼

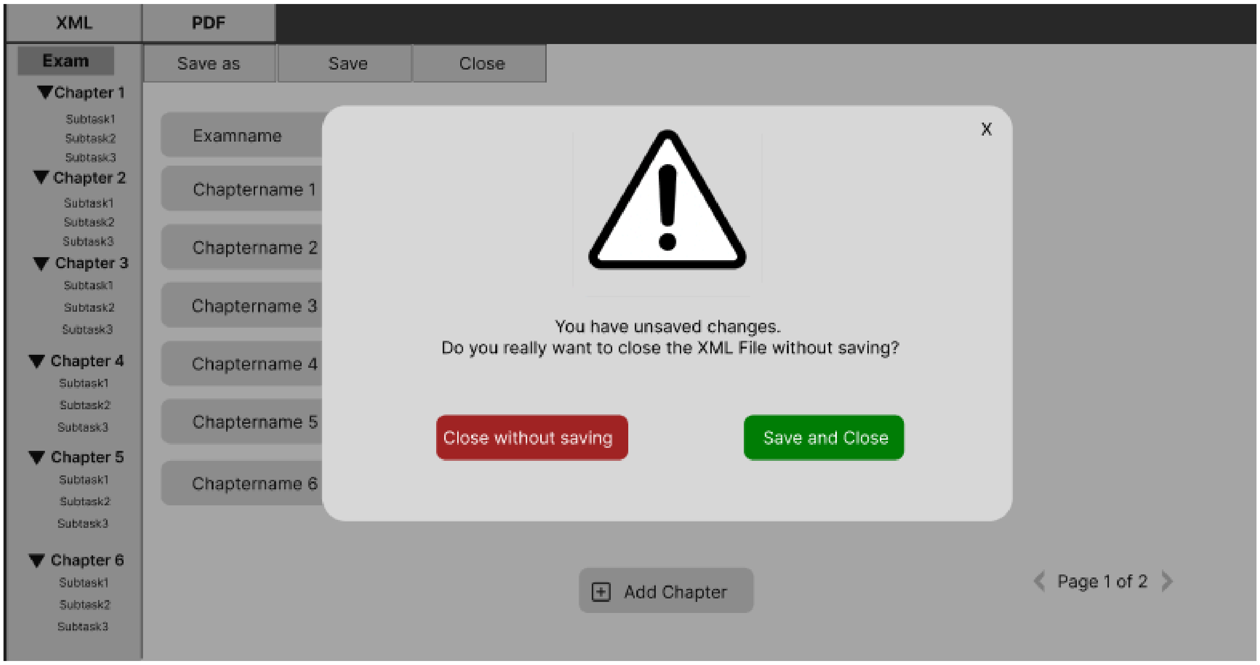
Chaptername 10 Score: 20 ▼ Exam Appearance: Include ▼

The picture above shows the screen when the user edits the score of a chapter. This is done by clicking on the arrow in the score field, which opens a dropdown menu. The dropdown menu displays any number in the selectable score range. When the user hovers over a specific number, its background is highlighted in blue color.



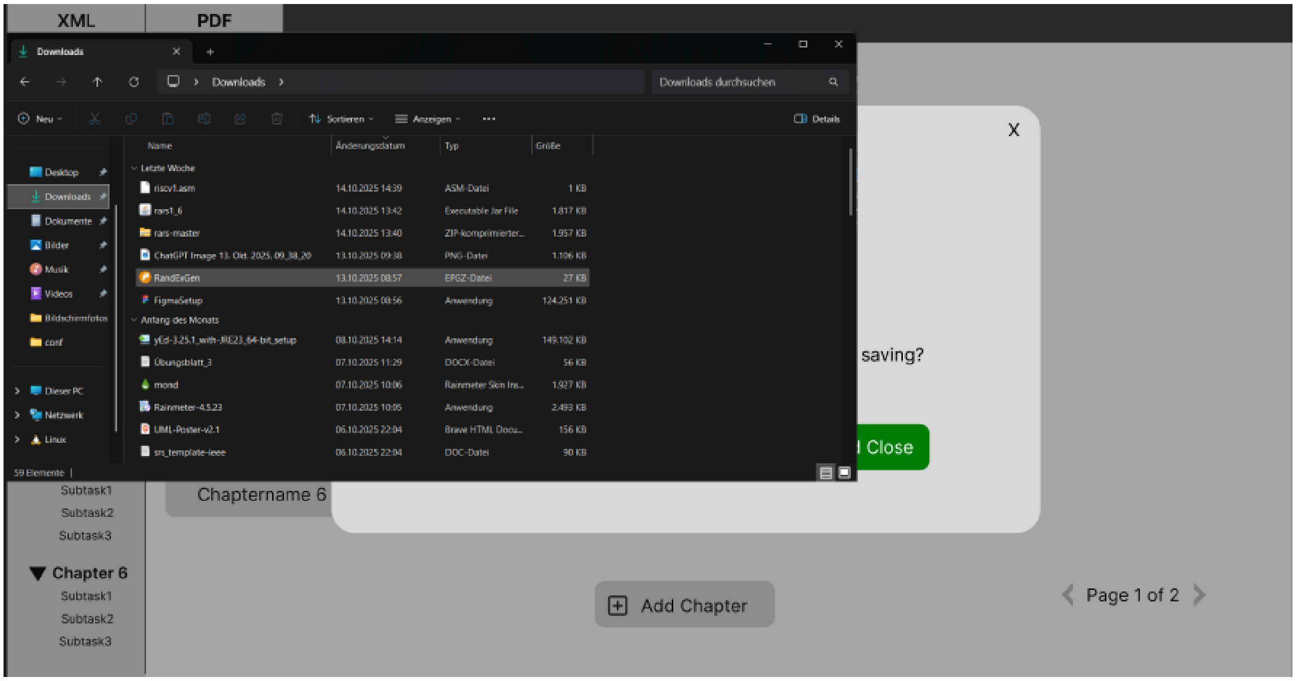
Name	Änderungsdatum	Typ	Größe
Letzte Woche			
riscv1.asm	14.10.2025 14:39	ASM-Datei	1 KB
rars1_6	14.10.2025 13:42	Executable Jar File	1.817 KB
rars-master	14.10.2025 13:40	ZIP-komprimierter...	1.957 KB
ChatGPT Image 13. Okt. 2025, 09_38_20	13.10.2025 09:38	PNG-Datei	1.106 KB
RandExGen	13.10.2025 08:57	EPGZ-Datei	27 KB
FigmaSetup	13.10.2025 08:56	Anwendung	124.251 KB
Anfang des Monats			
yEd-3.25.1_with-JRE23_64-bit_setup	08.10.2025 14:14	Anwendung	149.102 KB
Übungsblatt_3	07.10.2025 11:29	DOCX-Datei	56 KB
mond	07.10.2025 10:06	Rainmeter Skin Ins...	1.927 KB
Rainmeter-4.5.23	07.10.2025 10:05	Anwendung	2.493 KB
UML-Poster-v2.1	06.10.2025 22:04	Brave HTML Docu...	156 KB
srs_template-ieee	06.10.2025 22:04	DOC-Datei	90 KB

The screen above shows what happens when the user clicks the "Save as" button in the overview. An explorer opens in the top left corner, where the user can select the location and the name of the XML File. If the user saved the XML File once and wants to save new changes, you can just click the "Save" button. By clicking this button, the saved XML File is automatically overwritten in the background. If the user did not already save the file by clicking "Save as", the application will force the user to select a location and a file name in the explorer window. That's why this only happens when the user clicks on the "Save" button to save the XML file for the first time.



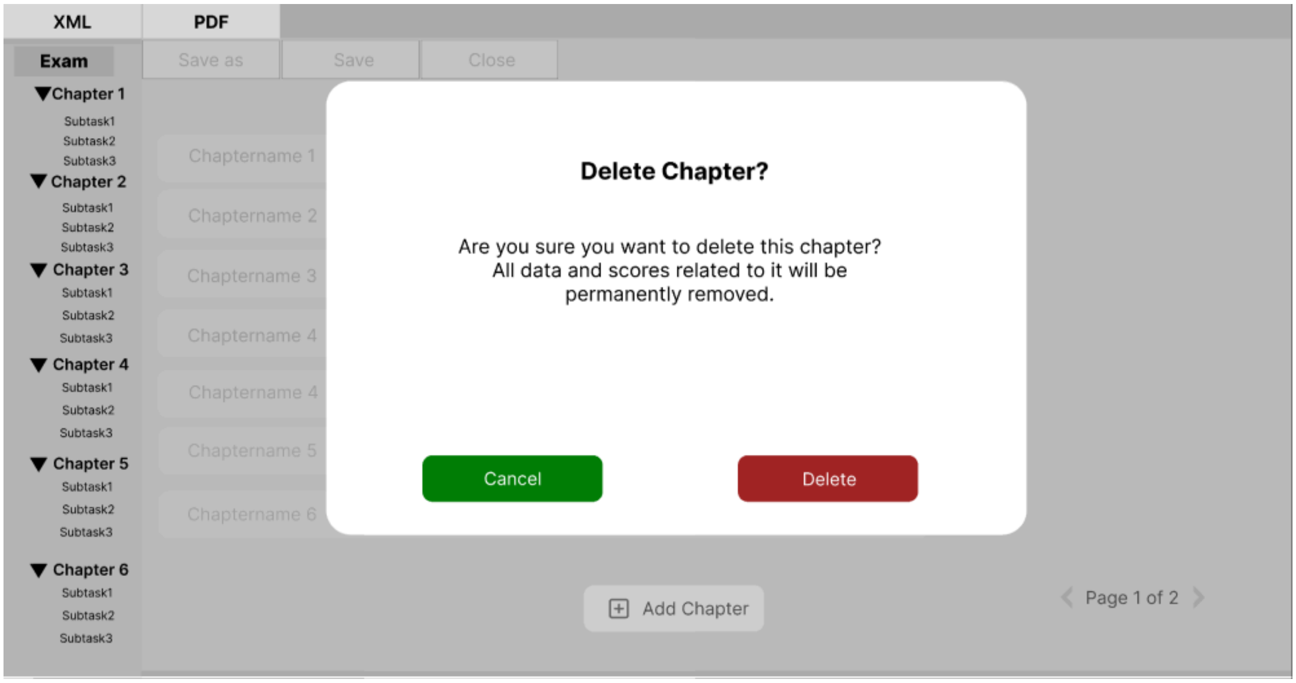
You have unsaved changes.
Do you really want to close the XML File without saving?

This screen above is shown if the user presses the “Close” button in the overview, when you have not saved the XML File yet. The application shows the warning message “You have unsaved changes. Do you really want to close the XML File without saving?”, to make sure that the user does not lose the XML File by accident. The user can either close the message by clicking on the “X” in the top right corner, which will take you to the overview again. Otherwise, the user can select between “Close without saving” and “Save and Close”. If the user selects “Close without saving”, you will be taken to the home screen again without the XML File being saved. If you click “Save and Close”, the user will be taken to the home screen after saving the XML File.



Name	Änderungsdatum	Typ	Größe
Letzte Woche			
riscv1.asm	14.10.2025 14:39	ASM-Datei	1 KB
rars1_6	14.10.2025 13:42	Executable Jar File	1.817 KB
rars-master	14.10.2025 13:40	ZIP-komprimierter...	1.957 KB
ChatGPT Image 13. Okt. 2025, 09_38_20	13.10.2025 09:38	PNG-Datei	1.106 KB
RandExGen	13.10.2025 08:57	EPGZ-Datei	27 KB
FigmaSetup	13.10.2025 08:56	Anwendung	124.251 KB
Anfang des Monats			
yEd-3.25.1_with-JRE21_64-bit_setup	08.10.2025 14:14	Anwendung	149.102 KB
Übungsblatt_3	07.10.2025 11:29	DOCX-Datei	56 KB
mond	07.10.2025 10:06	Rainmeter Skin Ins...	1.927 KB
Rainmeter-4.5.23	07.10.2025 10:05	Anwendung	2.493 KB
UML-Poster-v2.1	06.10.2025 22:04	Brave HTML Docu...	156 KB
srs_template-ieee	06.10.2025 22:04	DOC-Datei	90 KB

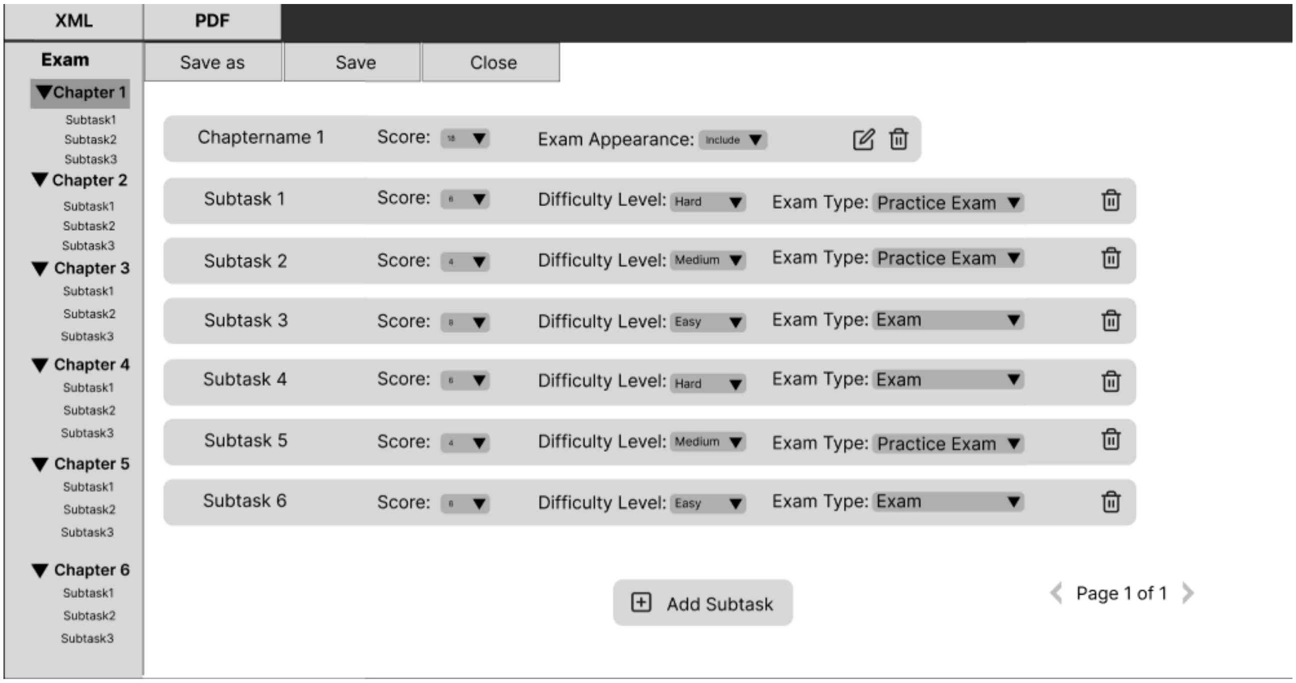
The screen above is shown when the user clicks the “Save and Close” button in the warning message. An explorer window opens in the top left corner, which is how you can select where to save the XML File.



Delete Chapter?

Are you sure you want to delete this chapter?
All data and scores related to it will be
permanently removed.

This screen above is shown if the user presses the trash bin symbol to remove a chapter. The application asks the user if he really wants to remove the chapter or if he wants to keep it. To remove the chapter, he can click on the red “Delete” button and if he wants to keep it, he can click on the green “Cancel” button.



Chaptername 1 Score: 18 ▼ Exam Appearance: Include ▼

Subtask 1 Score: 8 ▼ Difficulty Level: Hard ▼ Exam Type: Practice Exam ▼

Subtask 2 Score: 4 ▼ Difficulty Level: Medium ▼ Exam Type: Practice Exam ▼

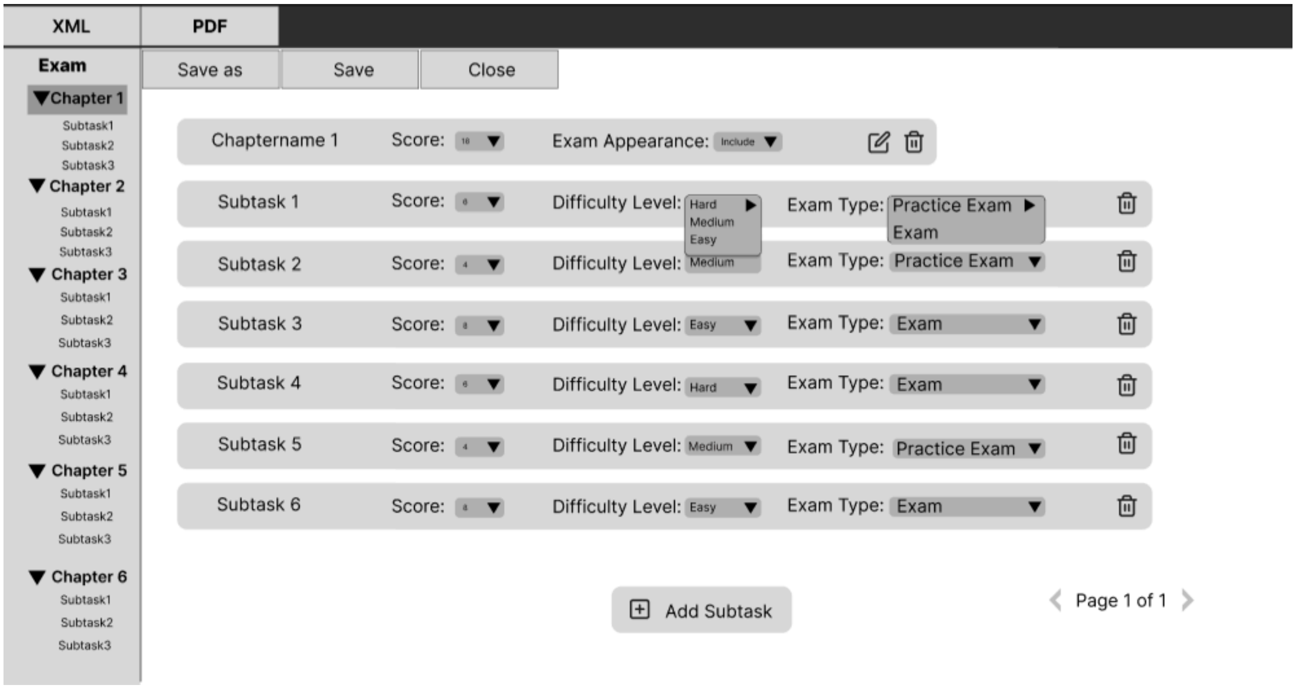
Subtask 3 Score: 8 ▼ Difficulty Level: Easy ▼ Exam Type: Exam ▼

Subtask 4 Score: 6 ▼ Difficulty Level: Hard ▼ Exam Type: Exam ▼

Subtask 5 Score: 4 ▼ Difficulty Level: Medium ▼ Exam Type: Practice Exam ▼

Subtask 6 Score: 8 ▼ Difficulty Level: Easy ▼ Exam Type: Exam ▼

The image above shows the screen when the user selected a chapter in the tree structure. It displays the name of the selected chapter, the score of the chapter, and the options to edit the title of the chapter or delete it. Also, the subtasks of the chapters are displayed, which can be deleted. In addition, the difficulty level of a subtask is shown. This is important, because a Chapter is only valid if it includes at least one subtask from every difficulty level (easy, medium, hard). The exam type is displayed too, as a subtask can be used for the generation of an exam or practice exam. The difficulty level and the exam type can also be changed by writing a new number in the score box. There is again an option in the bottom right corner to switch to the next page to see every subtask of a chapter. The user also has the possibility to add a subtask through the “Add Subtask” button at the bottom center.



Chaptername 1 Score: 18 Exam Appearance: Include

Subtask 1 Score: 6 Difficulty Level: Hard Exam Type: Practice Exam

Subtask 2 Score: 4 Difficulty Level: Medium Exam Type: Practice Exam

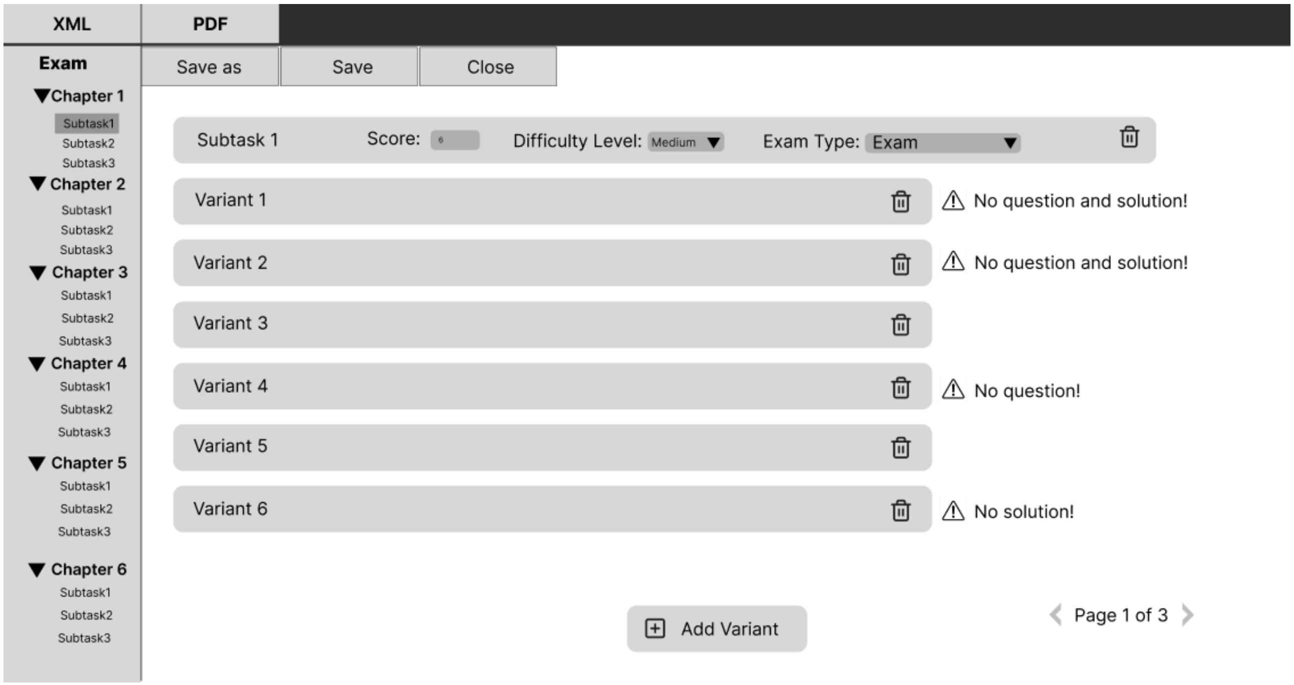
Subtask 3 Score: 8 Difficulty Level: Easy Exam Type: Exam

Subtask 4 Score: 6 Difficulty Level: Hard Exam Type: Exam

Subtask 5 Score: 4 Difficulty Level: Medium Exam Type: Practice Exam

Subtask 6 Score: 4 Difficulty Level: Easy Exam Type: Exam

The picture above shows the screen when the user edits the difficulty level and the exam type of a subtask. By clicking the arrow in the Difficulty Level and Exam Type field, you open the dropdown menu to edit the data. If the user hovers over a specific value of the data, the background will be highlighted in blue color.



Subtask 1 Score: 6 Difficulty Level: Medium Exam Type: Exam

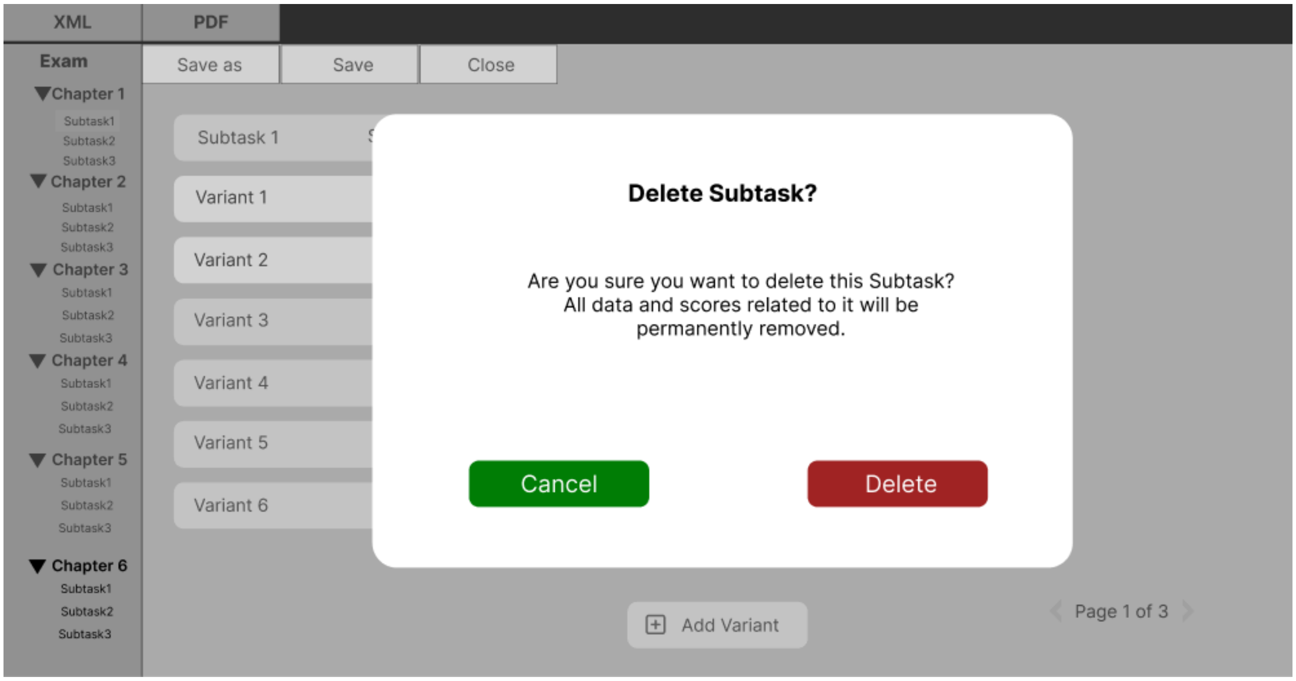
Variant 1 ⚠ No question and solution!

Variant 2 ⚠ No question and solution!

Variant 4 ⚠ No question!

Variant 6 ⚠ No solution!

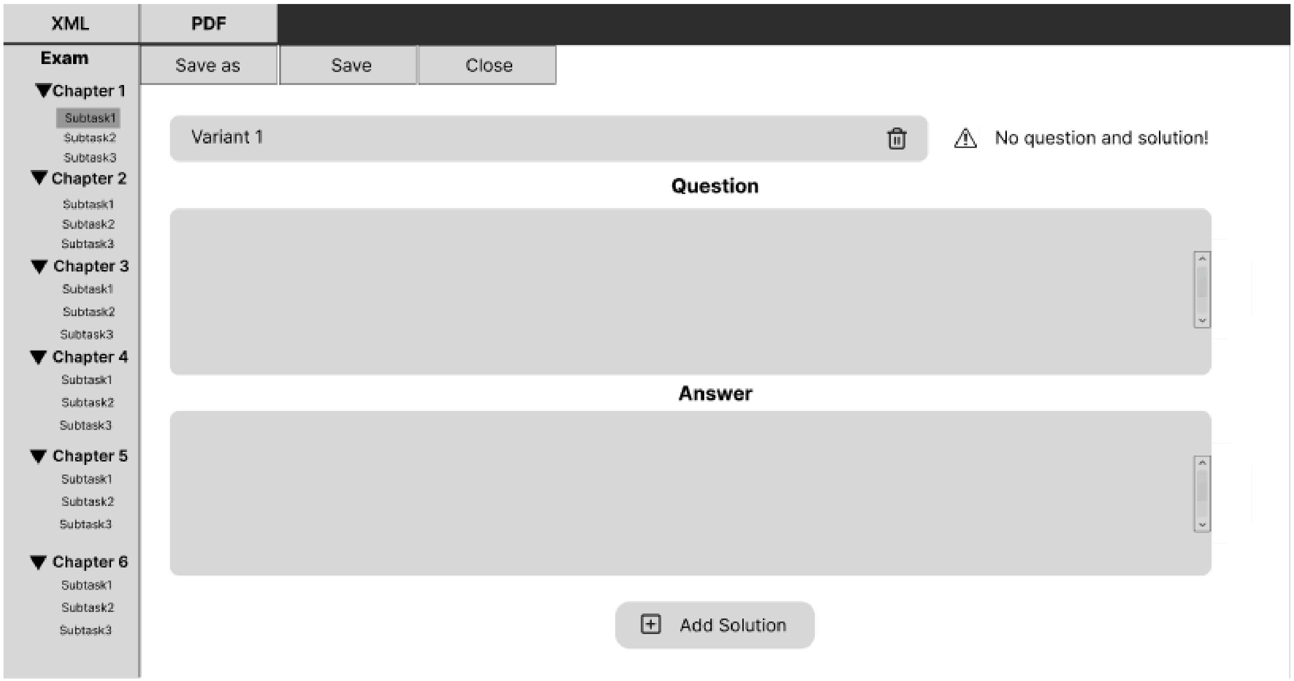
The picture above shows the screen when the user selects a Subtask in the tree structure. On the right side of the screen, the subtask and the subordinate variants are shown. In the bottom right corner, the user can again switch to the next page to see every variant of a subtask. The user has again the option to change the score, the difficulty level and the exam type of the selected subtask. In addition, the user can also delete entire subtasks, as well as add new variants to the subtask. When the user adds a variant, the application jumps to the last page to display it. Also, a variant is only valid if a question and solution is added, which is why an error message is shown on the right if it is invalid.



Delete Subtask?

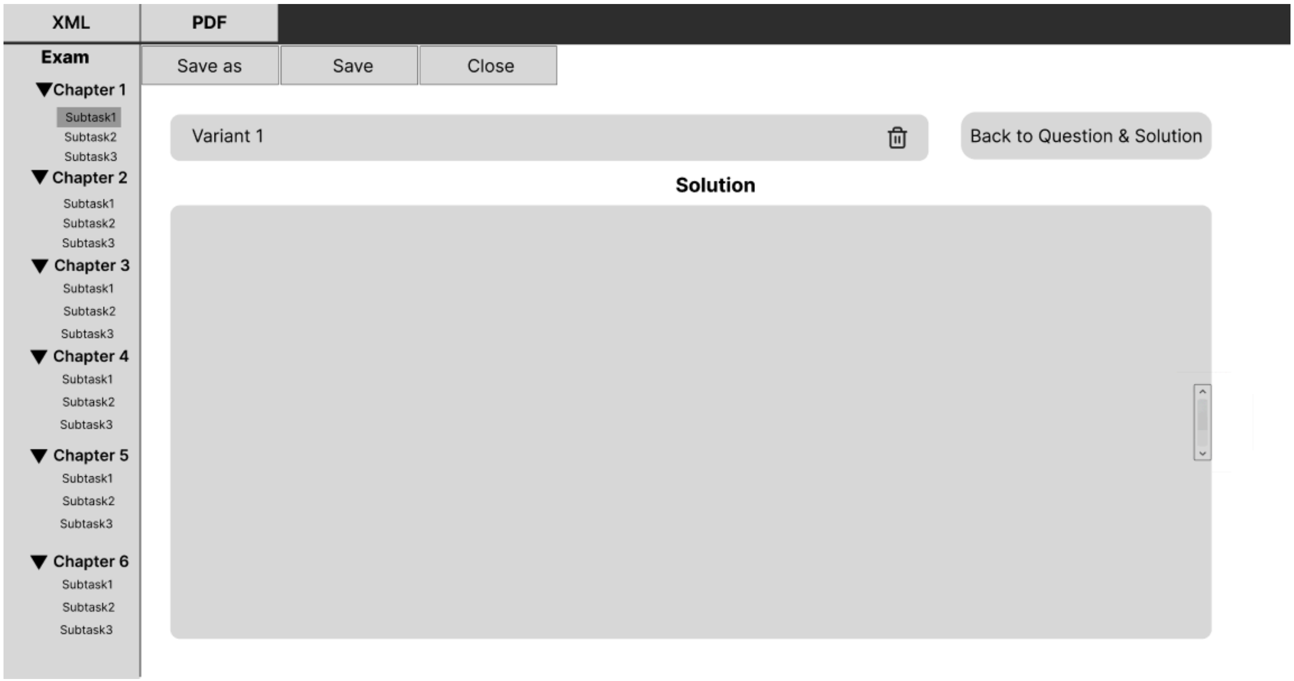
Are you sure you want to delete this Subtask?
All data and scores related to it will be
permanently removed.

The picture above shows the screen when the user decides to delete a subtask. A message is shown where the user must confirm the deletion of the subtask, to make sure that it does not get deleted by accident. This happens if the user clicks on the red “Delete” button. If the user wants to keep the subtask, the action can be canceled by clicking on the green “Cancel” button.

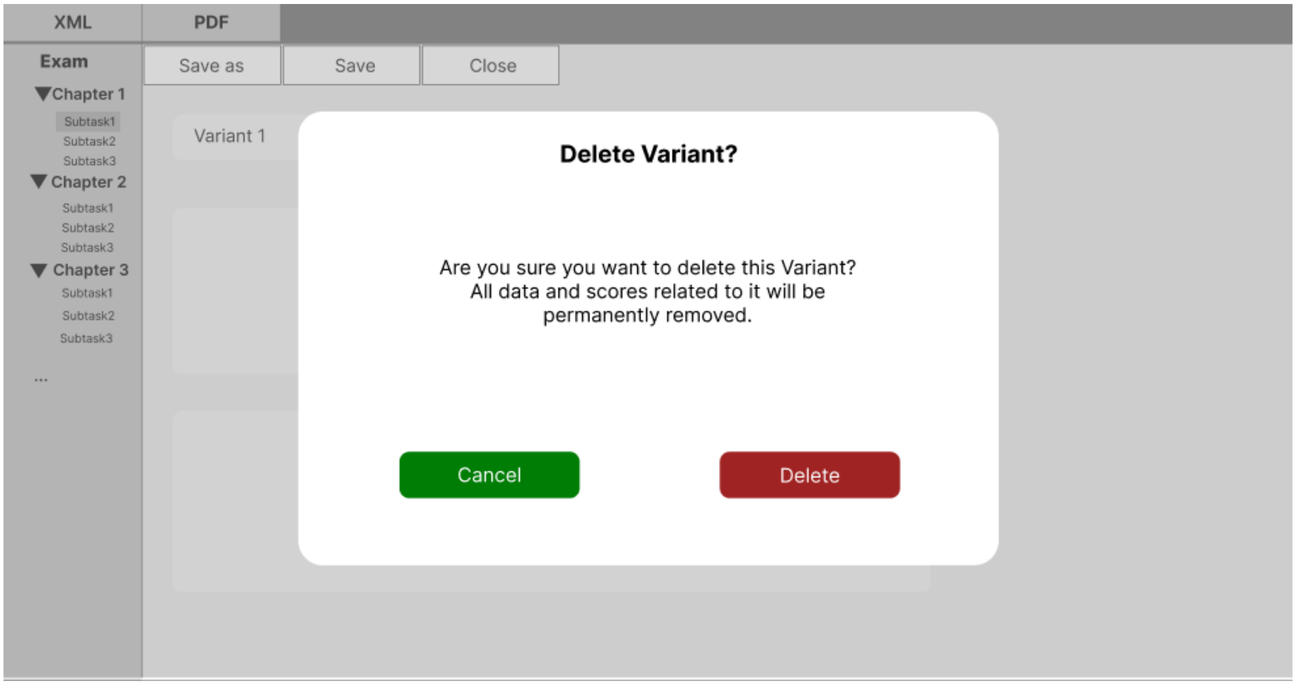


⚠ No question and solution!

If the user clicks on a subordinate variant of a subtask, the application shows the screen above. On the left side of the screen, the subtask which contains the selected variant is highlighted. On the right side of the screen, the user has the possibility to delete the entire variant. If the variant is invalid, the error message is also displayed in this screen on the right of the variant. In addition, a question field and an answer field are shown. The question field contains the question, which will be displayed in the final exam if this variant is chosen from the random generator. The answer field is often left empty but can also be edited if the user wants to provide an answer for this variant. This happens, for example, if the task is to correct the text in the answer field. The user can also edit the question and answer by clicking on their fields. If it is too long to display the full text, the user can use the scroll bar on the right side of the fields. Additionally, a solution must be added to validate the variant. The solution is necessary, because otherwise the PDF File which contains the solutions for the exam would be empty. To add a solution, the user can click on the “Add Solution” button in the bottom center.



The screen above is shown when the user clicks on the "Add Solution" button in the overview of a variant. The adding or editing of the solution happens by clicking on the solution field. If the text of the solution is too long, the user can again use the scroll bar on the right. If the user wants to go back to a subtask or chapter, the tree structure can be used as always. But if the user wants to go back to the overview of the selected variant, the button in the top right corner can be used.

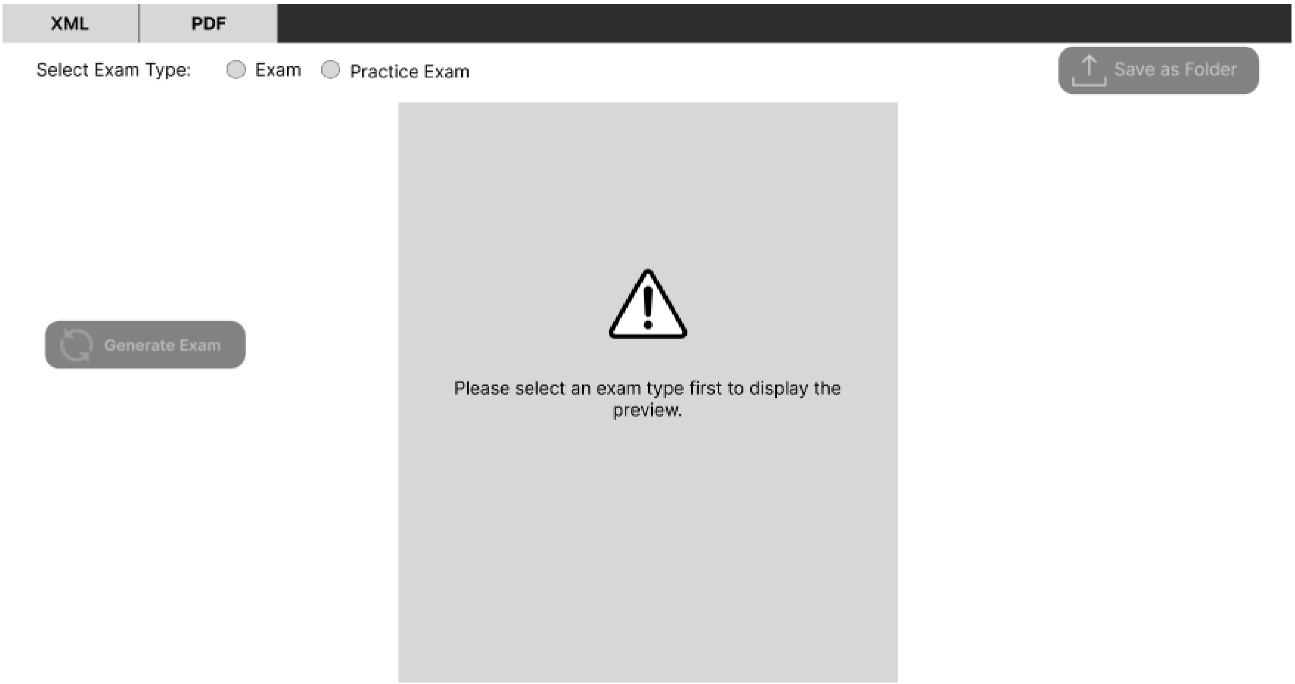


Delete Variant?

Are you sure you want to delete this Variant?
All data and scores related to it will be
permanently removed.

The picture above shows the screen when the user wants to delete a variant. It's the same process as deleting a chapter or a subtask. The application shows a message to the user, where the deletion can be confirmed by clicking on the red “Delete” button, or the deletion can be cancelled by clicking on the green “Cancel” button.

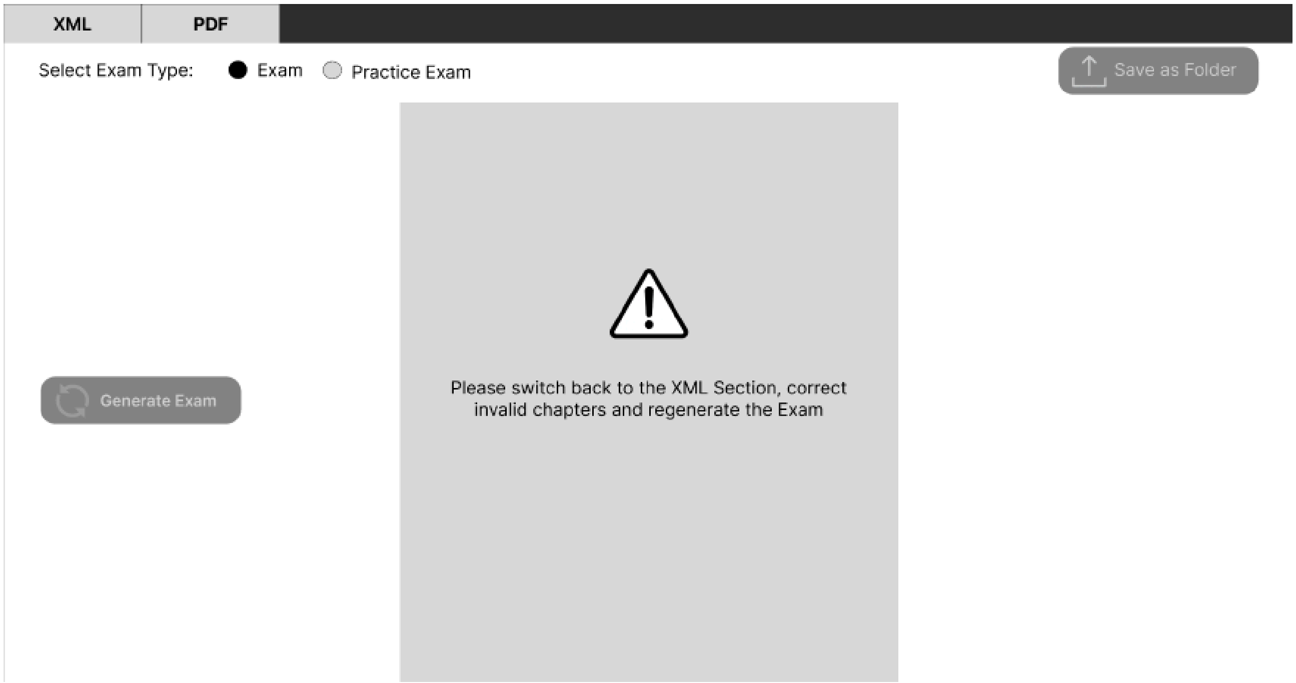
PDF Section



Please select an exam type first to display the preview.

The picture shows the screen when the user switches to the PDF Section of the application. In the top left corner, you can select the exam type. You can choose between an exam and practice exam, which will lead to only subtasks of the same exam type appearing in the exam. In the middle on the left there is a “Generate Exam” button, which is faded colored to show that the user cannot press it if no exam type is selected. In the middle center you can see a preview of the PDF File of the exam. As the selection of the exam type is necessary to display the PDF preview, there is an error message “Please select an exam type first to display the preview”, if the user did not select one.

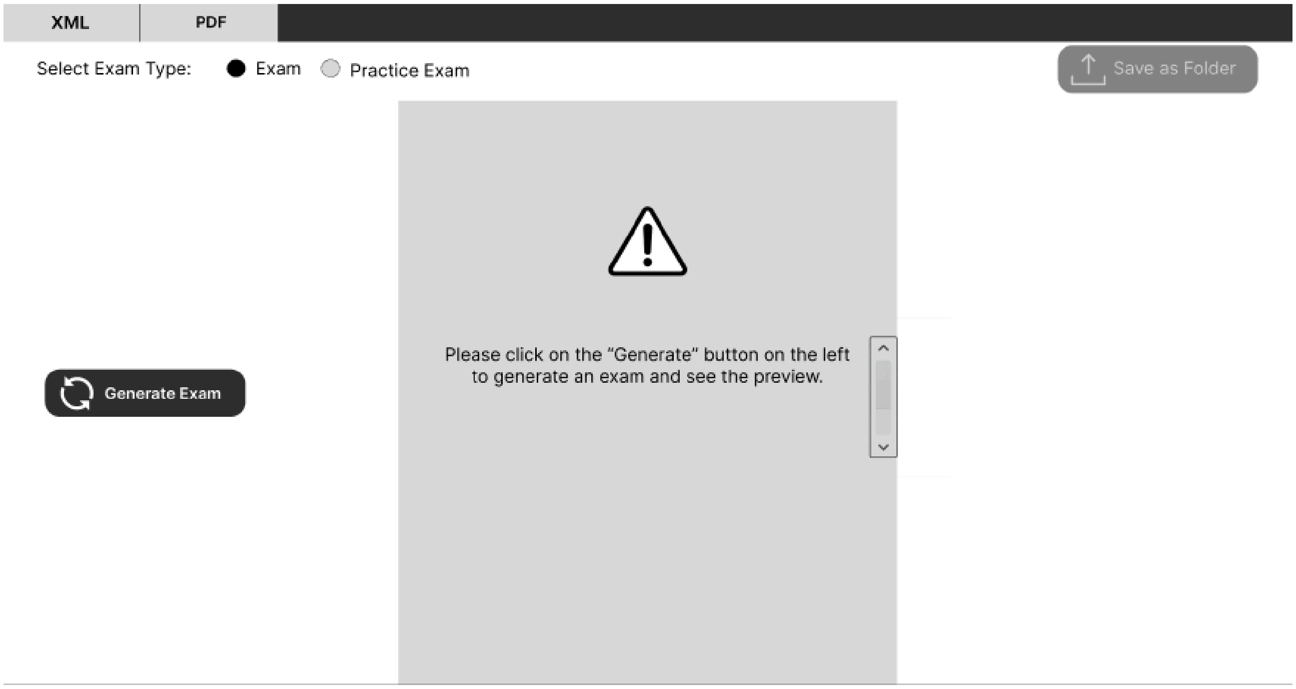
In the top right corner, there is the button “Save as Folder”, because not only will the exam itself be saved as PDF, also the solutions of the exam. That’s why the user saves a Folder on the system, which includes the exam, as well as its solutions. This button is also faded colored, because an exam can only be saved on the system, if it is already generated.



Select Exam Type: ● Exam ○ Practice Exam

Please switch back to the XML Section, correct invalid chapters and regenerate the Exam

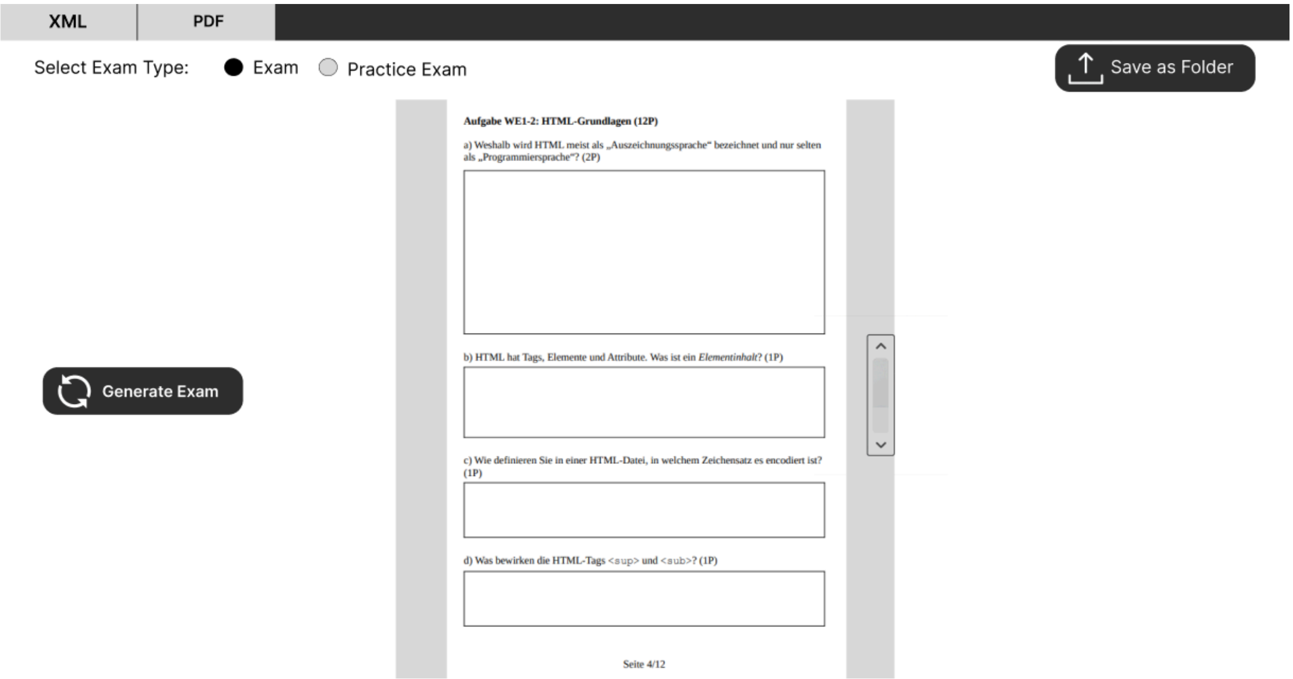
The picture above the screen when the user selected a regular exam as the exam type. In this state, the generation of the exam is not possible because the user has invalid chapters in the XML File. This is why the “Generate Exam” and “Save as Folder” buttons are faded colored. Instead of the preview, there is the message “Please switch back to the XML Section, correct invalid chapters and regenerate the exam”. When the user now goes back to the XML Section, the chapters can be corrected to unlock the exam generation, exam preview, and the final export.



Select Exam Type: ● Exam ○ Practice Exam

Please click on the "Generate" button on the left to generate an exam and see the preview.

The picture above shows the screen when the user selected an exam type but did not click the "Generate Exam" button. Instead of the preview, there is the message "Please click on the Generate button on the left to generate an exam and see the preview". By clicking on the "Generate Exam" button, the generation process is performed, which leads to the preview and the export being unlocked.



Select Exam Type: ● Exam ○ Practice Exam

Aufgabe WE1-2: HTML-Grundlagen (12P)

a) Weshalb wird HTML meist als „Auszeichnungssprache" bezeichnet und nur selten als „Programmiersprache"? (2P)

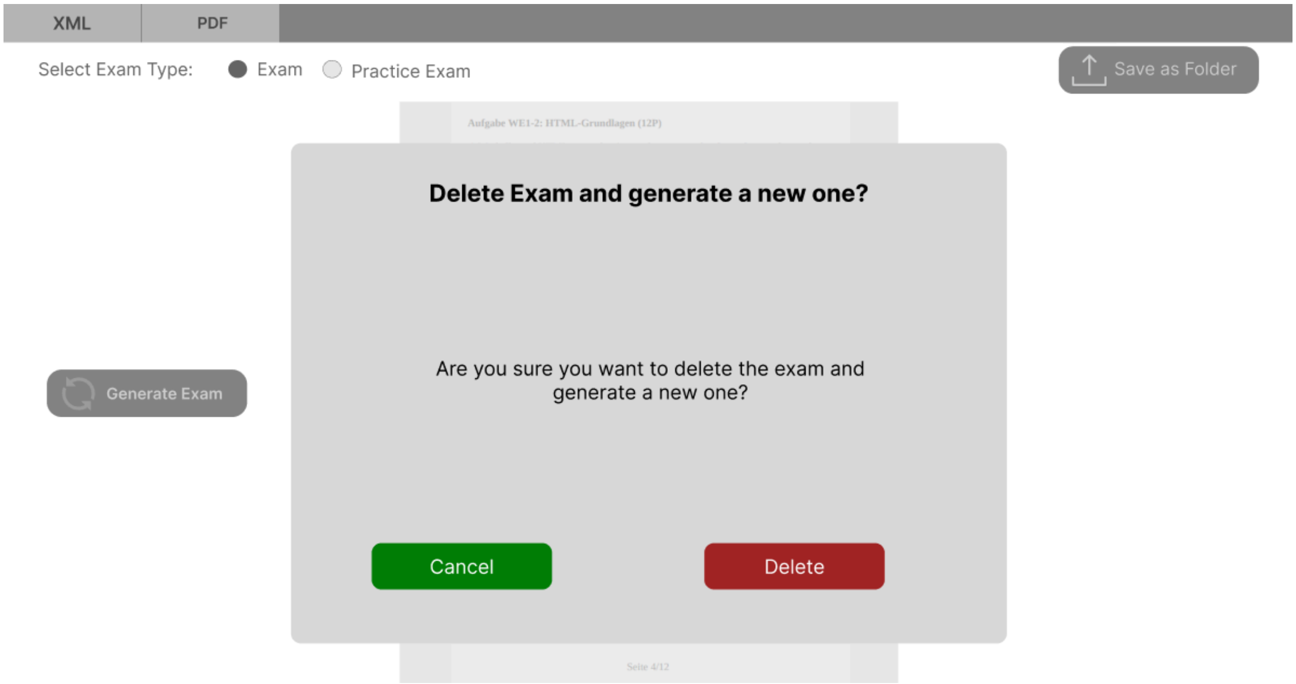
b) HTML hat Tags, Elemente und Attribute. Was ist ein *Elementinhalt*? (1P)

c) Wie definieren Sie in einer HTML-Datei, in welchem Zeichensatz es encodiert ist? (1P)

d) Was bewirken die HTML-Tags <sup> und <sub>? (1P)

Seite 4/12

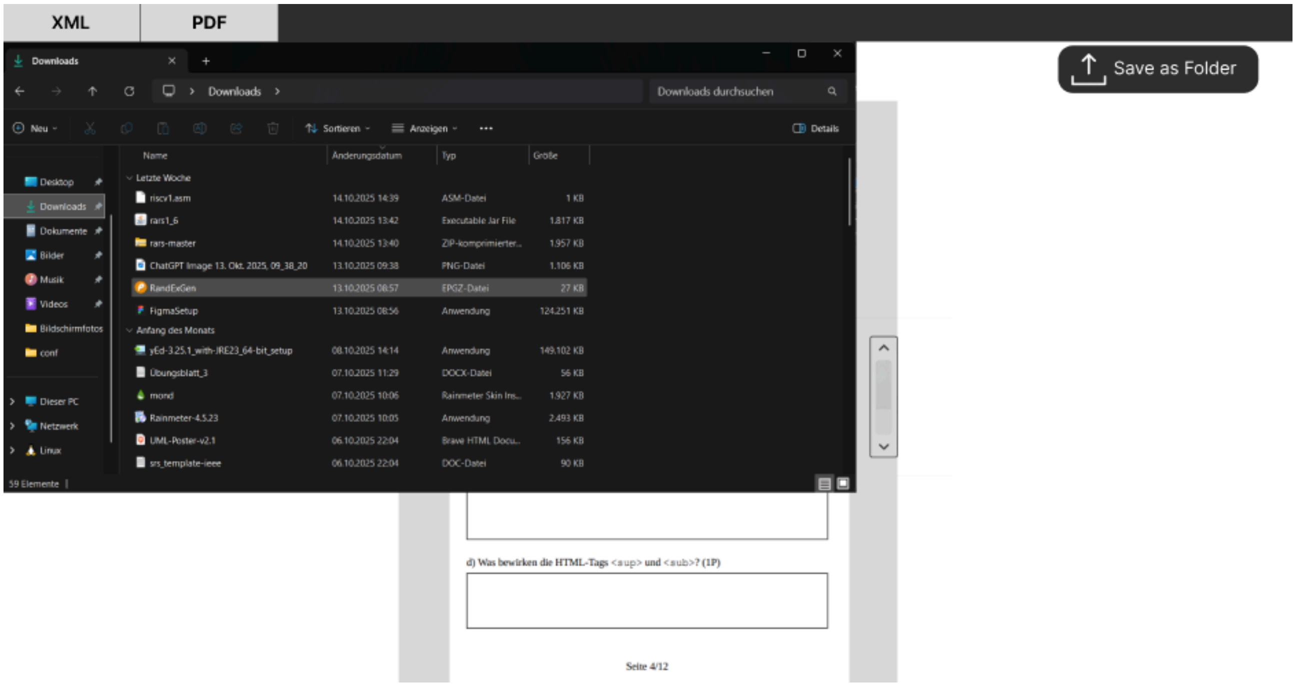
This screen shows the PDF Section when the user selected an Exam type and clicks on the "Generate Exam" button. In the middle of the screen, a preview with a scrollbar on the right can be seen. The scrollbar makes sure that the user can easily switch between pages of the previewed PDF document. If the user wants to save the File on the system, this can be done by clicking on the “Save as Folder” button. There “Generate Exam” button on the left can be clicked to delete this exam and generate a new one.



Delete Exam and generate a new one?

Are you sure you want to delete the exam and generate a new one?

The picture above shows the screen when the user already generated an exam and clicks on the “Generate Exam” button again. The user can perform this action to generate a new exam of it does not meet the expectations. The shown error message and the buttons to confirm or cancel the deletion ensure that the exam is not deleted by accident.



Name	Änderungsdatum	Typ	Größe
Letzte Woche			
riscv1.asm	14.10.2025 14:39	ASM-Datei	1 KB
rars1_6	14.10.2025 13:42	Executable Jar File	1.817 KB
rars-master	14.10.2025 13:40	ZIP-komprimierter...	1.957 KB
ChatGPT Image 13. Okt. 2025, 09_38_20	13.10.2025 09:38	PNG-Datei	1.106 KB
RandExGen	13.10.2025 08:57	EPGZ-Datei	27 KB
FigmaSetup	13.10.2025 08:56	Anwendung	124.251 KB
Anfang des Monats			
yEd-3.25.1_with-JRE23_64-bit_setup	08.10.2025 14:14	Anwendung	149.102 KB
Übungsblatt_3	07.10.2025 11:29	DOCX-Datei	56 KB
mond	07.10.2025 10:06	Rainmeter Skin Ins...	1.927 KB
Rainmeter-4.5.23	07.10.2025 10:05	Anwendung	2.493 KB
UML-Poster-v2.1	06.10.2025 22:04	Brave HTML Docu...	156 KB
srs_template-ieee	06.10.2025 22:04	DOC-Datei	90 KB

d) Was bewirken die HTML-Tags <sup> und <sub>? (1P)

The picture above shows what happens when the user clicks the “Save as Folder” button. It will lead to an explorer window opening in the top left corner. Then the user can select the location where the Folder should be saved. The folder includes the generated exam, as well as the solutions, both as separate PDF Files.

4. System Requirements

4.1 Functional Requirements

The following functional requirements define the essential capabilities that the RandExGen software must provide.
They describe the expected behavior of the system from the user’s perspective and specify the core functionalities required to generate, validate, and export exams.
Each requirement has a user priority and a technical priority on a scale from 1 to 5.

4.1.1 FREQ-1: Exam types

The system must allow the user to generate both regular exams and practice exams based on the available XML dataset. The user must be able to specify the exam type (regular or practice) for each subtask. Depending on the selected type, the generation process will include or exclude the different subtasks.

User Priority: 5/5 - This feature represents the core functionality of the system and is essential for its intended academic use.

Technical Priority: 4/5 - The implementation requires controlled task filtering and XML attribute handling, but it can be realized using existing data structures and logic without external dependencies.

4.1.2 FREQ-2: Difficulty Levels

The system must ensure that each generated exam has a balanced distribution of difficulties in every chapter.
Each chapter consists of an equal number of easy, medium, and hard subtasks. This is why a chapter must contain at least one subtask from each difficulty level.
If an even difficulty split is not possible, the system still accepts a small difference of two in the difficulty level. If the difference is higher than two or there is not a subtask from each difficulty level, the system prevents the generation process and highlights the invalid chapter.

User Priority: 4/5 - Maintaining a balanced difficulty level across all chapters is important for a balanced exam generation.

Technical Priority: 5/5 - This requirement directly impacts the random generation algorithm and must be enforced through data validation and logical constraints before exam generation.

4.1.3 FREQ-3: Randomized Generation

The system must generate exams and practice exams using a randomized selection algorithm that ensures unique combinations of tasks and subtasks each time the user executes a generation. Every subtask has the same chance to appear in the exam, which is why it's completely random-

User Priority: 5/5 - The randomization feature represents a major functionality of RandExGen and provides the focus to the user by offering variety and unpredictability in generated exams.

Technical Priority: 5/5 - Implementing randomness is essential for the system’s correctness and reliability. It requires handling of data structures, logic, and validation of errors.

4.1.4 FREQ-4: Loading and Creation of XML Files

The system must allow users to create a new exam dataset in XML and to load existing XML files. Users can edit exam data, overwrite existing files and save them as completely new XML Files (Save / Save As).

User Priority: 5/5 - The loading of existing XML data and the creation of new files is necessary to provide an enjoyable user experience.

Technical Priority: 5/5 - As the whole exam generation is based on the XML data, the creation and loading of XML Files is necessary for the system.

4.1.5 FREQ-5: Exam & Exam Solution Export

The system must give the user the possibility to export the generated Exam and Solutions as two separate PDF Files. The user must choose the output folder and filename. The user includes both PDF Files.

User Priority: 5/5 - Delivering printable exam and solution documents is essential for real-world use.

Technical Priority: 4/5 - The system could technically function without the export feature, but since it represents the purpose of the application, it is technically essential to the system’s functionality.

4.1.6 FREQ-6: Exam Preview

The system may provide an in-application PDF preview of the generated exam before exporting. The preview reflects the final layout and must represent proportions correctly. Users can navigate through the pages and refresh the preview after changing settings or dataset edits. Preview is only available after the creation or import of a complete XML File, otherwise there will be an error message instead of the preview.
User Priority: 4/5 - The system gives the user access to a preview before the final export of the PDF File, which is why the user can validate the final document without saving it on the system.

Technical Priority: 1/5 - The system is completely functional without a PDF preview, which is why this requirement is only relevant to increase the user experience.

4.1.7 FREQ-7: Exam Appearance

The system must give the user the possibility to choose which chapters in the XML File will be included or excluded from the exam generation. The user must include at least 4 valid chapters in the exam generation. The maximum number of chapters which can be included in the exam generation are 9 chapters.

User Priority: 5/5 - The system gives the user the possibility to include or exclude chapters from the exam generation, which is how the user has complete control about which chapters appear in the exam.

Technical Priority: 5/5 - The selection of the exam appearance of chapters is essential for the final document.

4.1.8 FREQ-8: Score

The system must give the user the possibility to freely choose the score of each subtask. Also, the score of chapters must be editable but not completely free. The user can select between a specific range of numbers. The lowest score limit of the chapter is the total score of the three subtasks with the lowest score from the three different difficulty levels (Subtask with the lowest score from difficulty level easy + Subtask with the lowest score from difficulty level medium + Subtask with the lowest score from difficulty level hard). The highest score limit of the chapter is the combination of the subtasks with the highest scores. The combination depends on the rule that subtasks from each difficulty level are 1/3 of the whole chapter with a maximum difference of two.

User-Priority: 5/5 - It is essential for the user to choose the score for each chapter and subtask to have complete control over the generated exam.

Technical Priority: 5/5 - The system must contain an algorithm that calculates every single combination of subtask by respecting the rules. Otherwise, the whole randomized generation process is not functional.

4.1.9 FREQ-9: Chapter Validation

The system must recognize if a chapter is invalid for a certain exam type. A chapter is only valid if it contains at least one subtask for each difficulty level of both exam types. If a chapter only contains enough subtasks with each difficulty level of a regular exam, it is invalid for the generation of a practice exam. Also, if a chapter only contains enough subtasks with each difficulty level of a practice exam, it is invalid for the generation of a regular exam. In addition, if a chapter does not contain enough subtasks with each difficulty level from both exam types, it is invalid for the generation of both exam types. If a chapter is invalid and included in the exam generation, the generation must be prevented. Also, the user must receive a notification.

User-Priority: 5/5 - It is essential for the user to only be able to generate Exams and Practice Exams with a valid dataset. Otherwise, there will be incorrect data in the generated exam.

Technical Priority: 5/5 - The system must contain an algorithm that checks the validation status of every chapter. Also, the generation must be prevented if there are invalid chapters included in the exam generation, to make sure that there is no incorrect data in the final exam.

4.1.10 FREQ-10: Subtask & Variant Validation

As the system must recognize if a chapter is invalid for the generation of an exam, it must also recognize if the contained data (Subtasks and Variants) is valid. A subtask can only be valid and included into the exam generation, if it contains at least one valid variant. This is necessary, because if a subtask does not contain any valid variant, no content of this subtask can be displayed in the exam. To recognize if a variant is valid, the system checks its' question and solution. The question of a variant must be at least 10 signs long to count as valid. This ensures that every question on the generated exam is clear enough to achieve a full score in this subtask. The solution of a variant must be at least 5 signs long to ensure that the generated PDF Solution File does not contain any errors. If a subtask does not contain any variant or every variant is invalid, the entire subtask counts as invalid and is not included in the exam generation. This can also lead to a whole chapter counting as invalid, which is why the user must receive a notification if a subtask or variant is invalid.

User-Priority: 5/5 - It is essential for the user to only be able to generate Exams and Practice Exams with a valid dataset. Otherwise, there will be incorrect data in the generated exam.

Technical Priority: 5/5 - The system must contain an algorithm that checks the validation status of every subtask and variant. If invalid subtasks or variants lead to invalid chapters which are included in the exam generation, it must be prevented to ensure that there is no incorrect data in the final exam.

4.1.11 FREQ-11: Exam Regeneration

Besides the normal generation process of an exam / practice exam, the system must give the user the possibility to generate a completely new exam if it does not meet his expectations.

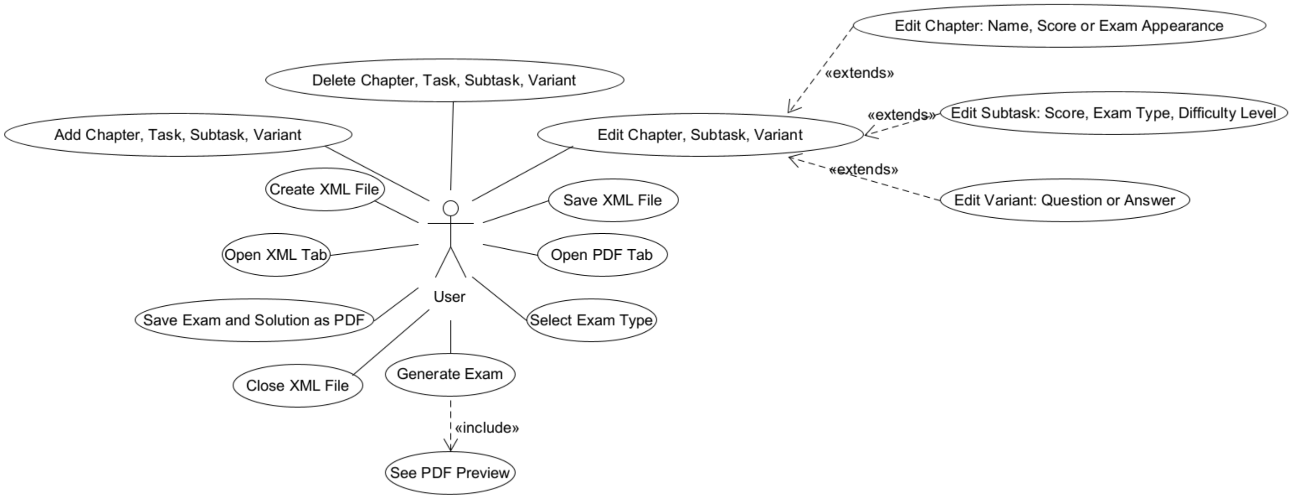
User Priority: 5/5 - It is essential that the user has complete control about which exam is generated and that an exam meets the users' expectations.

Technical Priority: 5/5 - The system must contain a function that lets the user delete the old exam and generate a new one. Otherwise, the generation process would not be user-friendly.

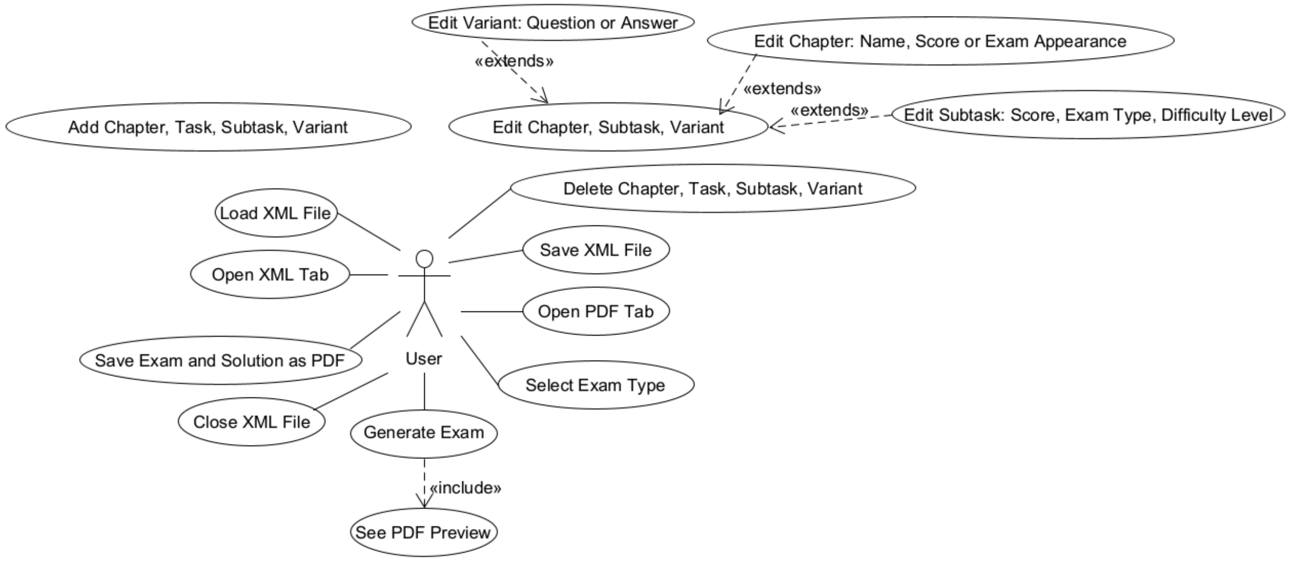
5. System Scenarios

5.1 Use-case Diagrams

Exam Generation by creating an XML File


Edit Chapter: Name, Score or Exam Appearance
Delete Chapter, Task, Subtask, Variant
«extends»
«extends»
Edit Subtask: Score, Exam Type, Difficulty Level
Add Chapter, Task, Subtask, Variant
Edit Chapter, Subtask, Variant
«extends»
Create XML File
Save XML File
Edit Variant: Question or Answer
Open XML Tab
Open PDF Tab
User
Save Exam and Solution as PDF
Select Exam Type
Close XML File
Generate Exam
«include»
See PDF Preview

Exam Generation by loading an XML File


Edit Variant: Question or Answer
Edit Chapter: Name, Score or Exam Appearance
«extends»
«extends»
«extends»
Add Chapter, Task, Subtask, Variant
Edit Chapter, Subtask, Variant
Edit Subtask: Score, Exam Type, Difficulty Level
Delete Chapter, Task, Subtask, Variant
Load XML File
Save XML File
Open XML Tab
Open PDF Tab
Save Exam and Solution as PDF
User
Select Exam Type
Close XML File
Generate Exam
«include»
See PDF Preview

5.2 Scenarios

The RandExGen software operates under two main scenarios, as illustrated in the *System Scenarios → Use-Case Diagrams* section.
The first scenario represents the generation of an exam based on an existing XML File, while the second scenario covers the creation of a completely new XML dataset from within the application.

Both scenarios start in the XML tab of the user interface. In this view, the user cannot access the PDF tab until a valid XML file has been either loaded or created. If the user attempts to open the PDF tab without doing so, the system displays an error message indicating that an XML file must be available first.

The difference between the two scenarios lies in step two:

- In the first scenario, the user loads an existing XML file from the file system.
- In the second scenario, the user creates a new XML dataset directly in the application.

Once the XML file is loaded or created, the user can manage the exam data. This includes:

- Adding or deleting chapters, subtasks, and variants.
- Editing existing data, such as chapter titles, subtask difficulty, score values, exam type, and exam appearance.
- Modifying the question text and model answers of variants.

After editing, the XML dataset can be saved to the local system.
The user can then switch to the PDF tab, where the system requires selecting an exam type (regular or practice exam) before displaying a preview of the generated exam in PDF format.

If further modifications are required, the user can return to the XML tab, edit the dataset, and the PDF preview updates automatically.
Finally, the user can export the results, saving a folder that contains both the exam and its solution as two separate PDF files.

5.2.1 SCN-1: Loading of an existing XML File for Exam generation

Name (ID)	Load XML (3)
Precondition	P1 The application is started
Main Flow	N1 XML Tab automatically opens as the starter screen N2 User clicks “Load XML File” button N3 Explorer window appears N4 User searches for location of XML-File in the explorer N5 User selects XML-File N6 XML File is loaded into the program
Postcondition	A Chapter overview of the XML Content is displayed
Alternative Flow	-
Errors	F1 Selected File is not of type XML F2 Selected File contains errors and therefore cannot be loaded
References	Open XML-Tab

5.2.2 SCN-2: Creation of a new XML File for Exam generation

Name (ID)	Create XML (2)
Precondition	P1 The application is started
Main Flow	N1 XML Tab automatically opens as the starter screen N2 User clicks “Create XML File” button N3 The XML File is created
Postcondition	A fresh Chapter overview is displayed
Alternative Flow	-
Errors	-
References	Open XML Tab

5.2.3 Common steps for SCN-1 and SCN-2

1. <u>XML Use-Cases</u>

Name (ID)	Open XML Tab (1)
Precondition	P1 The application is started
Main Flow	N1 User starts the application N2 User automatically opens XML-Tab as starting screen
Postcondition	XML-Tab is opened, and Overview of XML Content is displayed
Alternative Flow	N1 User is in the PDF-Tab N2 User hovers over to the Tab Section N3 User clicks on “XML” button
Errors	-
References	-

Name (ID)	Save XML File (4)
Precondition	XML-File was created / loaded
Main Flow	N1 User clicks “Save as” button N2 Explorer window appears

	N3 User selects file name N3 User selects file location N4 User saves XML-File on the system
Postcondition	XML-File is saved on the system with the selected name and location
Alternative Flow	
Errors	-
References	Open XML Tab

Name (ID)	Add Chapter, Subtask, Variant (5)
Precondition	P1 XML-Tab is opened P2 Superordinated exam component is displayed (1. Exam -> 2. Chapter -> 3. Subtask -> 4. Variant)
Main Flow	N1 User opens XML-Tab N2 User clicks on superordinated exam component of the desired component to be added N3 User clicks “Add Chapter”, “Add Subtask” or “Add Variant”
Postcondition	The desired exam component is added to XML-Content.
Alternative Flow	-
Errors	-
References	Open XML-Tab

Name (ID)	Delete Chapter, Subtask, Variant (6)
Precondition	P1 XML-Tab is opened P2 Desired Exam component is displayed (Chapter, Subtask or Variant)
Main Flow	N1 User opens XML-Tab N2 User navigates to desired component N3 User clicks on trash bin symbol
Postcondition	The desired exam component is deleted from XML-Content.
Alternative Flow	-
Errors	-
References	Open XML-Tab

Name (ID)	Edit Chapter (7)
Precondition	P1 XML- Tab is opened P2 Desired Chapter is displayed

Main Flow	N1 User opens XML-Tab N2 User navigates to desired chapter N3 User clicks on edit icon and selects new chapter name
Postcondition	XML-Content of the chapter is updated
Alternative Flow	N1 User opens XML-Tab N2 User navigates to desired chapter N3 User hovers over to dropdown menus and selects new score and exam appearance
Errors	-
References	Open XML Tab

Name (ID)	Edit Subtask (8)
Precondition	P1 XML-Tab is opened P2 Desired Subtask is displayed
Main Flow	N1 User opens XML-Tab N2 User clicks on superordinated chapter N3 User hovers over to desired subtask N4 User hovers over to dropdown menus and selects new score and difficulty level
Postcondition	XML-Content of the subtask is updated
Alternative Flow	N1 User opens XML-Tab N2 User clicks on superordinated chapter N3 User hovers over to desired subtask N4 User clicks on subtask N5 User hovers over to dropdown menus and selects new score and difficulty level

Errors	-
References	Open XML Tab

Name (ID)	Edit Variant (9)
Precondition	P1 XML-Tab is opened P2 Desired Variant is displayed
Main Flow	N1 User opens XML-Tab N2 User clicks on superordinated subtask N3 User hovers over to desired variant N4 User clicks on variant N5 User clicks on question or answer field N6 User updates text in question or answer field N7 User clicks “Enter” on keyboard
Postcondition	XML-Content of the variant is updated
Alternative Flow	-
Errors	-
References	Open XML tab,

2. PDF Use-Cases

Name (ID)	Open PDF –Tab (10)
Precondition	P1 An XML-File is loaded or created
Main Flow	N1 User is in the XML-Tab N2 User hovers over to the Tab Section N3 User clicks on “PDF” button
Postcondition	PDF-Tab is opened
Alternative Flow	-
Errors	F1 There is no XML loaded / created, the program shows
References	Open XML-Tab

Name (ID)	Select Exam Type (11)
Precondition	P1 PDF-Tab is opened

Main Flow	N1 User opens PDF-Tab N2 User hovers over to “Select Exam Type” Section N3 User chooses desired exam type by clicking on one of two boxes
Postcondition	If the user decides to click on “Generate Exam”, only subtasks from the selected exam type will be included.
Alternative Flow	-
Errors	-
References	Open PDF-Tab

Name (ID)	Generate Exam (12)
Precondition	P1 Exam Type is selected P2 Minimum of included chapters is fulfilled P3 All in the generation included chapters are valid
Main Flow	N1 User opens PDF-Tab N2 User selects an exam type N3 User clicks on “Generate Exam” button
Postcondition	Randomized exam out of included chapters is generated
Alternative Flow	-
Errors	F1 No exam type selected F2 Too few chapters are included F3 Included chapters are invalid
References	Open PDF-Tab; Select Exam Typ

Name (ID)	See PDF Preview (13)
Precondition	P1 Exam is generated
Main Flow	N1 User opens PDF-Tab N2 User selects exam type N3 User clicks “Generate Exam”
Postcondition	Preview of the generated exam can be seen in PDF-Tab
Alternative Flow	-
Errors	F1 Exam is not generated
References	Open PDF-Tab; Select Exam Type; Generate Exam

Name (ID)	Save Exam and Solution as PDF (14)
Precondition	P1 Exam is generated
Main Flow	N1 User clicks on “Save as Folder” button N2 User selects name of folder and location N3 User saves folder on the system
Postcondition	Folder is saved on the system with the selected name and location. It includes the generated exam and its solution as two separate PDF Files.
Alternative Flow	-
Errors	-
References	Generate Exam

6. System Constraints

6.1 Important Nonfunctional Requirements

The following non-functional requirements consist of quality and performance expectations for the application. Each requirement has a user priority and a technical priority on a scale from 1 to 5. These requirements should make sure that the software operates efficiently and provides a smooth user experience.

6.1.1 NFREQ-1: High usability

The system should offer a user-friendly, clear, and compact Graphical User Interface (GUI). The navigation through the application should be easy to understand for any person. Also, the user should get clear instructions, which leads to an easy exam generation process. All in all, the user must have an enjoyable and uncomplicated experience while using the software.

User Priority: 5/5 - A high usability is an extremely important requirement for the user. As the exam generation is fully done through the GUI, it is necessary to provide a user-friendly application.

Technical priority: 2/5 - As user-friendliness is important for improving the overall user experience, it is not essential for the functionality of the system. Therefore, it has a lower technical priority.

6.1.2 NFREQ-2: System Stability and Performance

The system should have excellent stability and performance during every process.
Functions, such as XML import, data edit, exam generation and PDF export must be executed smoothly without too long noticeable delay or crashes. The longest noticeable delay is ten seconds.

User Priority: 5/5 - A stable and fluid user experience is essential for efficient exam creation and professional usability.

Technical Priority: 4/5 - Achieving consistent performance requires optimized data handling, proper memory management, and efficient algorithm design.

7. Glossary

Score
Defines how many points you can score on the exam, chapters and subtasks
Given in form of a positive number or zero with the unit “P”
Can you only get even points?
Exam, Chapter, Subtask

Variant
Defines a version of a subtask.
Given in form of a text. It describes what you have to do to achieve a full score in this subtask.
There are several variants of a subtask, but only one variant is shown in the exam.
Chapter, Subtask

Chapter
Defines the topic and the module of following subtasks
Given in form of a heading
Contains at least one subtask from every difficulty level. If there are more than 3 subtasks, the number of subtasks per difficulty level must be equal. If an equal split is not possible, a difference of two is acceptable. As a chapter must include at least one subtask from every difficulty level, there have to be at least 3 subtask per chapter.
Exam, Subtask

Subtask
Defines the topic, score and difficulty level of subordinate variants. Also, it defines whether the subordinate variants are included or excluded in/from the exam generation.
Given in form of a text.
A subtask can contain several variants. The subtask is not directly visible in the exam, only one of its variants is visible.
Chapter, variant

Difficulty level
Defines how hard it is to solve a subtask
Given in form of a text: easy, medium or hard
Only subtasks have a certain difficulty level
Subtask

Exam
Defines the collection of all chapters
The exam is generated completely randomized.
Random generator

Random generator
To provide the generation of different exams, the chance for a subtask and its variants to appear in the exam is completely random.
The random generator still must respect all rules: A chapter must contain at least 3 subtasks from the different difficulty levels. If there are more than 3 subtasks, the split in terms of difficulty level must be equal. Only a small difference of 2 is acceptable.
Difficulty level, Exam, Chapter, Subtask, Variant

Exam data / Exam component
Expression for variants, subtasks, chapters and the whole exam with its dataset
Exam, Chapter, Subtask, Variant