

Software Engineering

Prüfungsleistung: Projekt

Version: 1.0

Moritz Köhler

December 7, 2025

Contents

1 Einleitung	2
1.1 Zweck des Dokuments	2
1.2 Überblick über das System	2
1.3 Hauptmerkmale	2
1.4 Systemumgebung	2
1.5 Nicht-funktionale Anforderungen	2
1.5.1 Leistungsanforderungen	2
1.5.2 Wartbarkeit	3
1.5.3 Benutzerfreundlichkeit	3
1.5.4 Sicherheit und Datenintegrität	3
1.5.5 Abhängigkeiten	3
2 Begriffslexikon	4
2.1 Klausur (Exam)	4
2.2 Kapitel (Chapter)	4
2.3 Aufgabe (Task)	4
2.4 Variante (Variant)	4
2.5 Punktwert / Scoring	4
2.6 PDF-Datei / Export	4
2.7 Validierung	5
2.8 Standardwerte	5
2.9 Strukturdigramm	5
3 Anwenderfälle	6
3.1 Akteure	6
3.2 Übersicht der Anwenderfälle	7
3.3 Anwenderfälle im Detail	7
4 Benutzeroberfläche	11
4.1 Navigation	11
4.2 Komponenten	11
4.3 Exam ausgewählt	12
4.4 Chapter ausgewählt	13
4.5 Task ausgewählt	13
4.6 PopUp	14
4.7 Rearrange Chapters	14

1 Einleitung

1.1 Zweck des Dokuments

Dieses Dokument beschreibt die Software **JExam**, ein Werkzeug zur Verwaltung von Aufgabenpools und zur automatisierten Generierung von Klausuren.

Es richtet sich an folgende Zielgruppen:

- **Entwickler:** Verantwortlich für Implementierung, Wartung und Erweiterung der Software.
- **Tester:** Leiten das korrekte Verhalten der Software anhand der Spezifikation ab.
- **Lehrende:** Möchten die Funktionsweise nachvollziehen und Aufgabenpools erstellen.

Darüber hinaus soll das Dokument die Benutzung, Architektur und nicht-funktionale Anforderungen der Software klar und nachvollziehbar darstellen.

1.2 Überblick über das System

JExam ist eine JavaFX-basierte Desktop-Anwendung zur Verwaltung von Aufgabenpools, aus denen automatisch Klausuren generiert werden können.

Die Aufgabenpools werden in **XML-Dateien** gespeichert und bestehen aus folgenden hierarchischen Elementen:

- **Kapitel / Chapter:** Oberkategorien für Aufgaben.
- **Aufgaben / Tasks:** Konkrete Aufgaben mit Punktzahl, Schwierigkeitsgrad und Sichtbarkeit.
- **Varianten / Variants:** Unterschiedliche Ausprägungen einer Teilaufgabe inklusive Musterlösung.

Beim Laden der XML-Dateien prüft das System die Struktur und Inhalte, um fehlerhafte Daten zu erkennen. Eine PDF-Generierung kann erst nach dieser Validierung und durch einen separaten Nutzerbefehl ausgelöst werden, wodurch sichergestellt wird, dass nur gültige Aufgaben exportiert werden.

1.3 Hauptmerkmale

- Import und Export von Aufgabenpools als XML-Dateien
- Verwaltung von Kapiteln (Chapter) und Aufgaben (Tasks)
- Verwaltung von Varianten mit unterschiedlichen Aufgabenstellungen und Lösungen
- Automatische PDF-Generierung mit oder ohne Musterlösung
- Unterstützung von Probeklausuren mit spezifisch gekennzeichneten tasks
- Mehrsprachige Benutzeroberfläche (ab Version 2)

1.4 Systemumgebung

JExam ist plattformübergreifend lauffähig auf:

- Windows 11 oder neuer
- macOS Monterey oder neuer
- Linux-Distributionen mit installiertem OpenJDK 17+

Die Software wird als ausführbares JAR-Paket verteilt und kann durch Doppelklick oder über die Konsole gestartet werden. Es sind Schreibrechte für XML- und PDF-Dateien erforderlich.

1.5 Nicht-funktionale Anforderungen

1.5.1 Leistungsanforderungen

- Die Benutzeroberfläche reagiert innerhalb von 500 ms auf Eingaben.
- Laden einer typischen XML-Datei (<1MB, 500 Subtasks) dauert maximal 2 s.

-
- Generierung einer PDF-Datei erfolgt in weniger als 5 s.
 - Skalierbarkeit: Aufgabenpools mit mehreren tausend Subtasks sollten ohne Absturz verarbeitet werden können.

1.5.2 Wartbarkeit

- Die Software folgt der MVC-Architektur (Model-View-Controller).
- Öffentliche Klassen und Algorithmen sind mit JavaDoc dokumentiert.
- Erweiterungen oder Optimierungen der Software können modular umgesetzt werden.
- Automatisierte Tests (Unit-Tests) unterstützen die Stabilität bei Änderungen.

1.5.3 Benutzerfreundlichkeit

- Intuitive Benutzeroberfläche mit Menüs, Dialogen und Übersichtslisten.
- Mehrsprachigkeit: UI kann auf alternative Sprachen umgestellt werden. (ab Version 2)

1.5.4 Sicherheit und Datenintegrität

- Lokale Speicherung: Alle XML- und PDF-Dateien werden auf dem Rechner des Nutzers verwaltet.
- Validierung der XML-Dateien beim Laden, um fehlerhafte Strukturen zu verhindern.
- Schreibrechte für Speicherverzeichnisse müssen vorhanden sein.
- Keine Netzwerkverbindung erforderlich, wodurch externe Sicherheitsrisiken minimiert werden.

1.5.5 Abhängigkeiten

- Java 17 oder höher
- JavaFX für die Benutzeroberfläche
- Externe Bibliotheken für PDF-Erzeugung und XML-Verarbeitung

2 Begriffslexikon

2.1 Klausur (Exam)

Wenn keine XML-Datei geladen ist, startet die Software mit einer Beispielstruktur, die vom Nutzer zur Erstellung der ersten oder weiteren XML-Dateien verwendet werden kann.

Name: New Exam

Diese enthält zum Start noch keine Kapitel/Aufgaben.

Das Exam dient als Grundlage um die Klausur zu generieren.

2.2 Kapitel (Chapter)

Ein Kapitel ist die übergeordnete Struktur, um Aufgaben einem Thema zuzuordnen. Der vergebene Name wird bei der Erstellung der PDF-Datei als Titel der Aufgabe verwendet.

Name: New Chapter

2.3 Aufgabe (Task)

Eine Aufgabe ist eine konkrete Aufgabenarchitektur und kann mehrere Varianten haben. Sie verwaltet zudem die zu erreichende Maximalpunktzahl, den Schwierigkeitsgrad und die Sichtbarkeit.

Name: New Subtask

Punktzahl: 1

Schwierigkeitsgrad: "easy"

Sichtbarkeit: "exam"

Variante: Startet mit einer Variante (siehe Variant)

Schwierigkeitsgrad:

- "easy"
- "medium"
- "hard"

Sichtbarkeit:

- "exam" - echte Klausur
- "mock-exam" - Probeklausur

2.4 Variante (Variant)

Eine Variante verwaltet die konkrete Aufgabenstellung mit einer möglichen Musterlösung. Diese werden direkt für das Erstellen der PDF-Datei verwendet.

Frage: New Question

Antwort: New Answer

2.5 Punktwert / Scoring

Jede Teilaufgabe hat eine maximale Punktzahl:

- Wird beim PDF-Export angezeigt.
- Dient bei der Klausurgenerierung zur Zielpunktsumme.

2.6 PDF-Datei / Export

Die Ausgabe der Klausur erfolgt als PDF.

- Klausur ohne Lösungen
- Musterlösung

-
- Probeklausur (Mock-Exam) ohne Lösungen
 - Probeklausur (Mock-Exam) mit Lösungen

2.7 Validierung

Prüft die Konsistenz des Exams:

- Jede Teilaufgabe muss mindestens eine Variante besitzen.
- Punktewerte, Schwierigkeitsgrad und Sichtbarkeit werden geprüft.
- Bei Fehlern wird die PDF-Erstellung blockiert.

2.8 Standardwerte

Für neu angelegte Elemente existieren Defaultwerte:

- Klausurname: "New Exam"
- Kapitelname: "New Chapter"
- Task Name: "New Subtask"
- Punkte: 1
- Schwierigkeitsgrad: "easy"
- Sichtbarkeit: "exam"
- Variante Frage/Antwort: "New Question" / "New Answer"

2.9 Strukturdiagramm

```
Exam (XML)
+-- Chapter
|   +-- Task
|   |   +-- Variant 1
|   |   +-- Variant 2
|   |   +-- ...
|   +-- ...
+-- ...
```

3 Anwenderfälle

3.1 Aktoren

Die primären Akteure der Software sind Dozierende oder anderes lehrendes Personal, welches Exams verwalten und daraus Klausuren generieren möchte. Es werden lediglich grundlegende Computerkenntnisse vorausgesetzt.

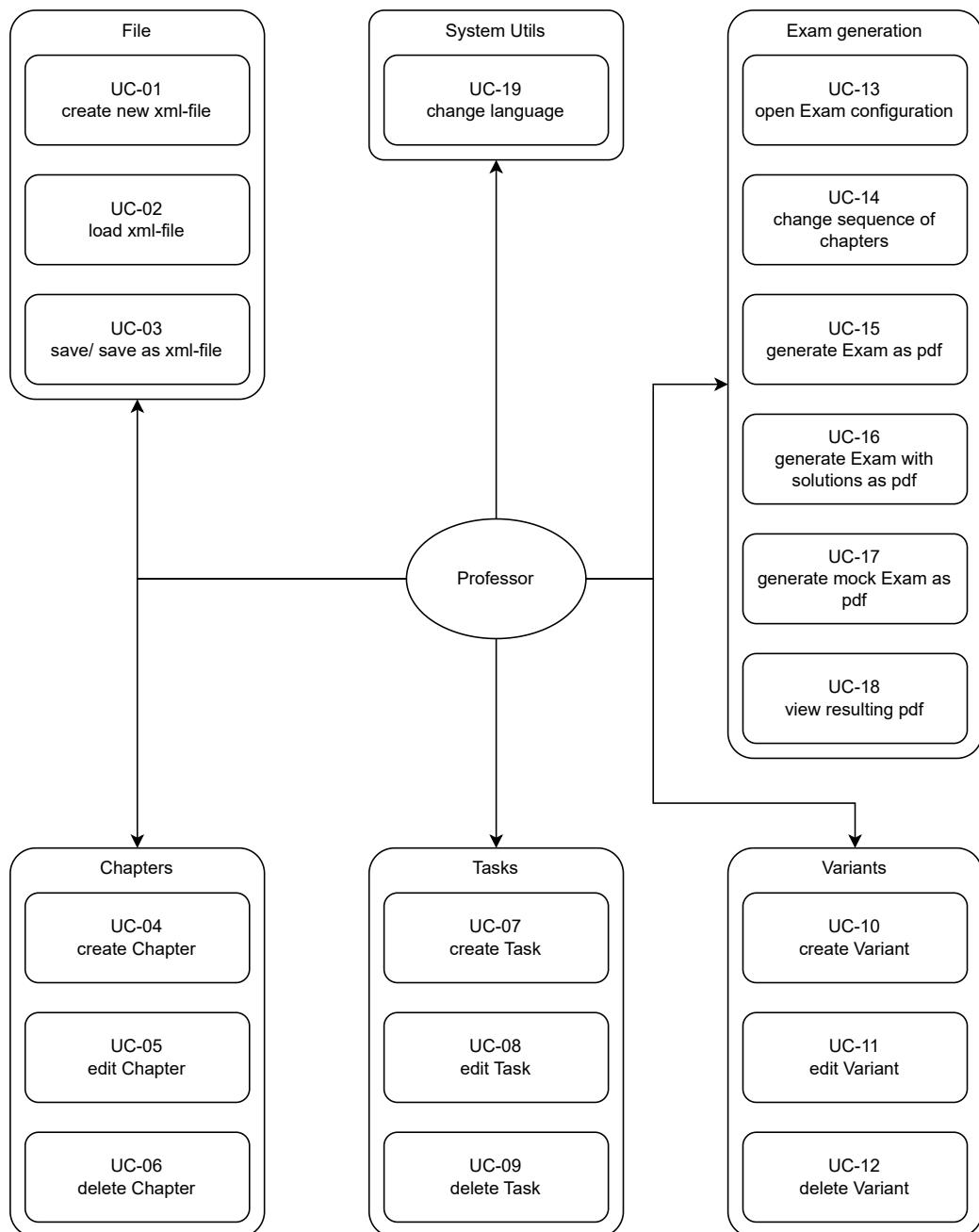


Figure 1: All Different Use Cases

3.2 Übersicht der Anwenderfälle

- UC-01 XML-Datei neu anlegen
- UC-02 XML-Datei laden
- UC-03 XML-Datei speichern / Speichern unter
- UC-04 Kapitel anlegen
- UC-05 Kapitel bearbeiten
- UC-06 Kapitel löschen
- UC-07 Aufgabe anlegen
- UC-08 Aufgabe bearbeiten
- UC-09 Aufgabe löschen
- UC-10 Variante anlegen
- UC-11 Variante bearbeiten
- UC-12 Variante löschen
- UC-13 Klausurkonfiguration öffnen
- UC-14 Reihenfolge der Kapitel ändern
- UC-15 Klausur generieren (PDF)
- UC-16 Musterlösung generieren (PDF)
- UC-17 Probeklausur generieren
- UC-18 PDF-Vorschau anzeigen
- UC-19 Sprache ändern

3.3 Anwenderfälle im Detail

UC-01: XML-Datei neu anlegen

Kurzbeschreibung:

Beim Start wird automatisch ein leerer Exam angelegt.

Vorbedingungen:

Keine XML-Datei muss geladen sein.

Ablauf:

1. Nutzer hat das Programm gestartet

Nachbedingungen:

Eine neue, leere Exam-Struktur ist im Speicher.

Fehlerbetrachtung: Nicht relevant (Operation kann nicht fehlschlagen).

UC-02: XML-Datei laden

Kurzbeschreibung:

Eine bestehende XML-Datei (Exam) wird geladen.

Vorbedingungen:

Datei muss existieren und korrektes Format besitzen.

Ablauf:

1. Nutzer wählt unter Menu "File" die Option "Open"
2. Nutzer wählt mittels Dateidialog eine XML-Datei.
3. System lädt und validiert die Struktur.

Nachbedingungen:

Der Aufgabenpool ist vollständig sichtbar.

Fehlerbetrachtung:

- Datei korrupt → Fehlermeldung, Abbruch.
- Datei hat falsche Struktur → Fehlermeldung, Abbruch.

UC-03: XML-Datei speichern / Speichern unter

Kurzbeschreibung:

Der aktuelle Aufgabenpool (Exam) wird gespeichert.

Vorbedingungen:

Ein gültiger Aufgabenpool wurde geladen oder erstellt.

Ablauf:

1. Nutzer wählt "Save" oder "Save as". Befindet sich unter Menu "File".
2. System schreibt XML-Datei.

Nachbedingungen:

Alle Daten sind persistent gespeichert.

Fehlerbetrachtung:

- Schreibfehler (z.B. fehlende Rechte) → Meldung.

UC-04: Kapitel anlegen

Kurzbeschreibung:

Ein neues Kapitel (Chapter) wird zum Aufgabenpool hinzugefügt.

Vorbedingungen:

Ein Aufgabenpool ist geladen oder neu erstellt.

Ablauf:

1. Nutzer klickt auf "New Chapter".
2. Kapitel wird mit Standardname angelegt.

Nachbedingungen:

Ein neues Kapitel existiert im Arbeitsspeicher

Fehlerbetrachtung: Keine - trivialer Vorgang.

UC-05: Kapitel bearbeiten

Kurzbeschreibung:

Kapitelname oder Metadaten eines Chapters werden geändert.

Vorbedingungen:

Kapitel existiert.

Ablauf:

1. Nutzer wählt Kapitel aus.
2. Nutzer ändert Namen und bestätigt mit "Save".

Nachbedingungen:

Kapitel wurde aktualisiert.

Fehlerbetrachtung: Keine - trivialer Vorgang.

UC-06: Kapitel löschen

Kurzbeschreibung:

Ein Kapitel (Chapter / Task) inklusive aller Unterelemente wird gelöscht.

Vorbedingungen:

Kapitel existiert.

Ablauf:

1. Nutzer klickt auf "Delete".
2. Sicherheitsabfrage erscheint.
3. Sicherheitsabfrage bestätigen.

Nachbedingungen:

Kapitel ist unwiderruflich entfernt.

Fehlerbetrachtung: Keine - außer Nutzer bricht ab.

UC-07: Aufgabe anlegen

Kurzbeschreibung:

Innerhalb eines Kapitels wird ein Task erstellt.

Vorbedingungen:

Mindestens ein Kapitel existiert.

Ablauf:

1. Nutzer wählt Kapitel.
2. Nutzer klickt "New Task".

Nachbedingungen:

Neuer Task mit Standardvariant existiert im Arbeitsspeicher

Fehlerbetrachtung: Keine.

UC-08: Aufgabe bearbeiten

Kurzbeschreibung:

Ein Task wird editiert.

Vorbedingungen:

Task existiert.

Ablauf:

1. Nutzer öffnet Task.
2. Nutzer ändert Punktzahl, Schwierigkeitsgrad, Sichtbarkeit.

Nachbedingungen:

Alle Änderungen sind im Arbeitsspeicher übernommen.

Fehlerbetrachtung:

- Punktzahl nicht im erlaubten Bereich.
- Schwierigkeitsgrad fehlt.
- Sichtbarkeit ungültig.

UC-09: Aufgabe löschen

Kurzbeschreibung:

Ein Task wird gelöscht.

Vorbedingungen:

Task existiert.

Ablauf:

1. Nutzer klickt auf "Delete".
2. Sicherheitsabfrage erscheint.
3. Sicherheitsabfrage bestätigen.

Nachbedingungen:

Task mit ist unwiderruflich gelöscht.

Fehlerbetrachtung: Keine - außer Nutzer bricht ab.

UC-10: Variante anlegen

Kurzbeschreibung:

Eine neue Variante wird einem Task hinzugefügt.

Vorbedingungen:

Task existiert.

Ablauf:

1. Nutzer klickt "Add Variant".

Nachbedingungen:

Variante existiert mit Standardwerten.

Fehlerbetrachtung: Keine.

UC-11: Variante bearbeiten

Kurzbeschreibung:

Frage- oder Antworttext einer Variante wird geändert.

Vorbedingungen:

Variante existiert.

Ablauf:

1. Nutzer öffnet Variante.
2. Nutzer bearbeitet Texte.

Nachbedingungen:

Änderungen sind im Arbeitsspeicher gespeichert.

Fehlerbetrachtung:

- Leere Frage → Fehlermeldung.

UC-12: Variante löschen

Kurzbeschreibung:

Eine Variante wird gelöscht.

Vorbedingungen:

Der Task hat mehr als eine Variante.

Ablauf:

1. Nutzer klickt auf "Delete".
2. Sicherheitsabfrage erscheint.
3. Sicherheitsabfrage bestätigen.

Nachbedingungen:

Variante gelöscht.

Fehlerbetrachtung: Keine.

UC-13: Klausurkonfiguration öffnen

Kurzbeschreibung:

Der Nutzer navigiert in den Tab zur Generierung einer Klausur (Exam).

Vorbedingungen:

Mindestens ein Kapitel mit gültigen Tasks existiert.

Nachbedingungen:

Konfigurationsdialog sichtbar.

Fehlerbetrachtung: Keine generierbaren Kapitel → Hinweis.

UC-14: Reihenfolge der Kapitel ändern

Kurzbeschreibung:

Der Nutzer ist in dem Tab zur Generierung einer Klausur (Exam)

Vorbedingungen:

Mindestens ein Kapitel mit gültigen Tasks existiert.

Ablauf:

1. Bewege die Kapitel mit den Buttons "UP" und "DOWN"
2. Wenn das Unterste Kapitel nach unten bewegt wird, so landet es im vom der Generierung ausgeschlossenem Bereich.

Nachbedingungen:

Reihenfolge und Ausschließung von Kapitel wird gewährleistet

UC-15: Klausur generieren (PDF)

Kurzbeschreibung:

System erstellt eine PDF-Klausur ohne Musterlösung aus dem Aufgabenpool (Exam).

Vorbedingungen:

- Klausur ist gültig (Difficulty 33%-Regel).
- Ausgabeort vom Nutzer gewählt.

Nachbedingungen:

PDF-Datei existiert.

Fehlerbetrachtung:

- Zielpunktzahl nicht möglich.
- Keine gültigen Aufgaben pro Kapitel.

UC-16: Musterlösung generieren

Kurzbeschreibung:

Erzeugt eine PDF-Datei mit eingebetteten Musterlösungen der Tasks/Variants.

Vorbedingungen:

Mindestens eine Variante hat eine Antwort.

Nachbedingungen:

”exam.lsg.pdf” existiert.

Fehlerbetrachtung:

- Keine Lösungen vorhanden → Warnung.

UC-17: Probeklausur generieren

Kurzbeschreibung:

Erzeugt eine rein aus Mock-Tasks bestehende Klausur.

Vorbedingungen:

Mindestens ein Task ist als ”mock-exam“ markiert.

Nachbedingungen:

Mock-Klausur-PDF existiert.

Fehlerbetrachtung:

- Keine Mock-Aufgaben vorhanden → Fehlermeldung.

UC-18: PDF-Vorschau anzeigen

Kurzbeschreibung:

Vor dem Export wird eine Vorschau der generierbaren Klausur (Exam) gerendert.

Vorbedingungen:

PDF muss generierbar sein.

Nachbedingungen:**Fehlerbetrachtung:**

- Fehlende PDF-Engine → Meldung.

UC-19: Sprache ändern

Kurzbeschreibung:

Die Benutzeroberfläche kann auf eine alternative Sprache umgestellt werden.

Vorbedingungen:

Alternative Sprache installiert.

Nachbedingungen:

UI lädt neue Übersetzungen.

Fehlerbetrachtung:

- Fehlende Übersetzung → Fallback zu Englisch.

4 Benutzeroberfläche

Die folgenden Abschnitte beschreiben die Benutzeroberfläche von **JExam** anhand der bereitgestellten UI-Bilder. Jede Abbildung wird durch einen erklärenden Text ergänzt, der beschreibt, welche Funktion dort ausgeführt wird und in welchem Kontext das Element genutzt wird.

Hinweis: Die Tatsächliche UI der Anwendung **JExam** kann leicht von der hier beschriebenen Benutzeroberfläche abweichen. Die Struktur und wo was ist bleiben Identisch.

4.1 Navigation

Header Navigation

Der Header ermöglicht das Umschalten zwischen der XML-Strukturansicht und der PDF-Konfiguration.



Figure 2: Navigation zwischen XML- und PDF-Modus.

Breadcrumbs

Die Breadcrumb-Leiste dient zur Navigation auf der aktuellen Hierarchieebene. Sie zeigt den Pfad zum aktuell ausgewählten Objekt (Exam → Chapter → Task).

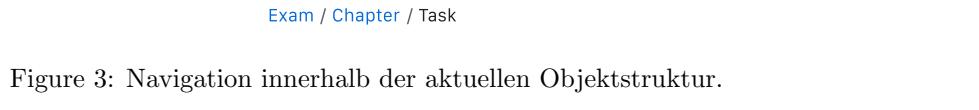


Figure 3: Navigation innerhalb der aktuellen Objektstruktur.

TreeView

Der TreeView zeigt die gesamte XML-Struktur (Exam → Chapter → Task). Über Dropdowns kann jede Ebene ein- und ausgeklappt werden.

Über Dropdowns kann jede Ebene ein- und ausgeklappt werden.
Ist noch kein Exam vorhanden, erscheint ein *Create Exam*-Button, der das entsprechende Popup öffnet.

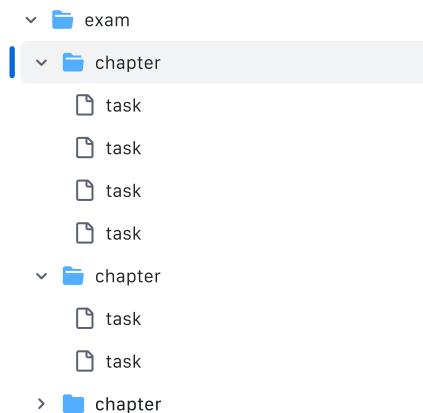


Figure 4: Navigation zwischen allen Objekten der XML-Datei.

4.2 Komponenten

Chapter Component

Eine Anzeige des Kapitels mit Buttons zum Bearbeiten oder Löschen. Löschen öffnet ein Popup und Bearbeiten navigiert in die Chapter Seite.



Figure 5: Chapter Component mit Edit- und Delete-Optionen.

Task Component

Ein Task wird mit Punktzahl, Schwierigkeit und Sichtbarkeit dargestellt. Enthält Edit- und Delete-Aktionen. Löschen öffnet ein Popup und Bearbeiten navigiert in die Chapter Seite.



Figure 6: Task Component mit Bearbeitungs- und Löschmöglichkeiten.

Variant Component

Zeigt Frage und Antwort einer Variante an und bietet Textfelder um die Fragen und Antworten zu bearbeiten und den Delete Button, welcher ein Popup öffnet.

A screenshot of a variant component. It shows two text input fields labeled "Question" and "Answer", both containing placeholder text "Example Question here" and "Example Answer here". To the right of each field is a small delete icon.

Figure 7: Variant Component zur Bearbeitung einzelner Varianten.

4.3 Exam ausgewählt

Exam Header

Ermöglicht das Umbenennen des Exams.

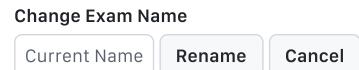


Figure 8: Exam Header zur Anpassung des Exam-Namens.

Chapter Table

Zeigt alle bestehenden Chapter Components an. Ein *Create Chapter*-Button öffnet das Kapitel-Erstellungs-Popup.

Name	Amounts			Questions per Difficulty				
Chapter Name	Tasks - 9	Variants - 15	Points - 31	Easy - 3	Medium - 5	Hard - 1		
Chapter Name	Tasks - 9	Variants - 15	Points - 31	Easy - 3	Medium - 5	Hard - 1		
Chapter Name	Tasks - 9	Variants - 15	Points - 31	Easy - 3	Medium - 5	Hard - 1		
Chapter Name	Tasks - 9	Variants - 15	Points - 31	Easy - 3	Medium - 5	Hard - 1		
New Chapter								

Figure 9: Tabelle aller Kapitel mit Option zur Neuerstellung.

4.4 Chapter ausgewählt

Chapter Header

Ermöglicht das Umbenennen sowie das Löschen eines Kapitels. Der Delete-Button öffnet die Bestätigungsabfrage.

Change Chapter Name

Figure 10: Header eines Kapitels mit Edit- und Delete-Funktionen.

Task Table

Die Tabelle zeigt alle Tasks des Kapitels. Mit dem Button *Create Task* wird das Task-Erstellungs-Popup geöffnet.

Name	Amounts	Scope	Difficulty	
Task Name	Variants - 2	Points - 5	Exam	Easy edit trash
Task Name	Variants - 2	Points - 5	Exam	Easy edit trash
Task Name	Variants - 2	Points - 5	Exam	Easy edit trash
Task Name	Variants - 2	Points - 5	Exam	Easy edit trash

Figure 11: Tabelle aller Tasks eines Kapitels.

4.5 Task ausgewählt

Task Header

Hier können die Eigenschaften des Tasks angepasst werden: Scope (*mock/exam*), Difficulty (*easy/medium/hard*) und Punktzahl. Der Delete-Button öffnet das Lösch-Popup.

Change Task Name

Select Scope
 Exam Mock

Select Difficulty
 Easy Medium Hard

Enter Points

Figure 12: Header eines Tasks mit konfigurierbaren Eigenschaften.

Variant List

Eine Liste aller Varianten des Tasks. Der *Create Variant*-Button öffnet das entsprechende Popup.

Question	<input type="button" value="Delete"/>
Example Question here	
Answer	<input type="button" value="Delete"/>
Example Answer here	
Question	<input type="button" value="Delete"/>
Second Question here	
Answer	<input type="button" value="Delete"/>
Second Answer here	
New Variant	

Figure 13: Variantenliste zur Verwaltung aller Varianten eines Tasks.

4.6 PopUp

Delete Popup

Popup zur Bestätigung des Löschens eines Class (ersetzen durch Chapter, Task, Variant). Gibt die Bedeutung zurück, dass auch alle Kindelemente unwiderruflich gelöscht werden.

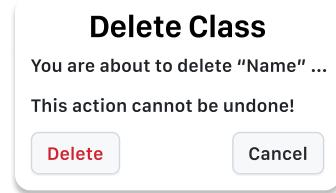


Figure 14: Löschbestätigung für Class

4.7 Rearrange Chapters

Rearrange Chapters & Create Exam

Ermöglicht die Reihenfolge der Chapter für die Klausurgenerierung zu ändern. Durch Verschieben in den "Excluded from PDF Generation" Bereich werden diese Chapter nicht mehr für die Generierung berücksichtigt und somit ausgeschlossen.

Change the order of the Chapters:

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Excluded from the PDF:

Chapter 5

Generate Exam

Generate Mock Exam

Figure 15: Umordnen der Chapters