

Author
Julian Garn
k11946733

Angefertigt am
**Institut für
Systemsoftware**

Beurteiler
Prof. Dr. **Hanspeter
Mössenböck**

Zweitbetreuer
Dr. **Markus Weninger**

November 2022

Web-Frontend für ein elektronisches Klausursystem



Bachelor Thesis
to obtain the academic degree of
Bachelor of Science
in the Bachelor's Program
Informatik

Kurzfassung

Da seit dem Semester 2020 viele Klausuren an der Johannes Kepler Universität Linz Covid-bedingt online stattgefunden haben, stieg das Interesse an Online-Klausursysteme deutlich. Das bisher eingesetzte System war allerdings nicht auf die hohe Anzahl an Studenten oder auf die Videoüberwachung während der Klausur ausgelegt. Wir haben daher ein System entwickelt, welches Prüfenden erlaubt Klausuren zu erstellen, zu verwalten und durchzuführen. Auch die Verwaltung der Teilnehmer ist ein wichtiger Teil des Projekts. Das System wurde mittlerweile mehrmals erfolgreich eingesetzt.

Abstract

Since 2020 many exams at the Johannes Kepler University Linz had to be held online due to Covid. Because of that, the interest in online exam programs has risen significantly. However, the system used so far was not designed for the high number of students or for video monitoring required during exams. We have therefore developed a system that allows examiners to create, manage and conduct exams. The administration of the participants is also an important part of the project. The system has been successfully used several times.

Inhaltsverzeichnis

1	Einführung	1
1.1	Bestehende Systeme	1
1.1.1	Moodle	2
2	Anforderungen	3
2.1	Anmeldung	3
2.2	Übersicht	4
2.2.1	Herunterladen und Entfernen	4
2.3	Bearbeiten	4
2.3.1	Elemente einer Klausur	5
2.3.2	Status und Vorschau	6
2.4	Teilnehmer verwalten	6
2.4.1	Gruppen	6
2.4.2	Import der Teilnehmer	7
2.4.3	Emails	8
2.4.4	Status	9
2.5	Übersicht für Aufsichtspersonen	9
2.5.1	Maßnahmen gegen Betrug	10
3	Datenstrukturen	11
3.1	Anmeldung	11
3.2	Übersicht	11
3.3	Bearbeiten	12
3.4	Teilnehmer verwalten	12
3.5	Übersicht für Aufsichtspersonen	12
4	Design	13
4.1	Anmeldung	13
4.2	Übersicht	14
4.3	Bearbeiten	14
4.3.1	Vorschau der Gesamtklausur	15
4.4	Teilnehmer verwalten	18
4.5	Übersicht für Aufsichtspersonen	20

Abbildungsverzeichnis

1.1	Eine Moodle Klausurfrage	2
4.1	Anmeldung	13
4.2	Übersicht	14
4.3	Die Eingabe der Grunddaten am oberen Ende der Oberfläche.	15
4.4	Die Eingabefelder für eine Überschrift sowie die Live-Vorschau.	15
4.5	Eingabefelder für eine Textfrage	16
4.6	Eingabefelder für eine Codefrage	16
4.7	Eingabefelder für eine single choice Frage	16
4.8	Eingabefelder für eine multiple choice Frage	17
4.9	Benutzeroberfläche unter den Klausursegmenten.	17
4.10	Vorschau	17
4.11	Verwaltung der Aufsichtspersonen	18
4.12	Verwaltung der Gruppen	19
4.13	Verwaltung der Email-Texte	19
4.14	Oberfläche für Aufsichtspersonen	20

1 Einführung

Durch die Verbreitung von COVID-19 wurde Anfang 2020 die Lehre an der Johannes Kepler Universität (JKU) soweit wie möglich auf Online- und Selbststudium umgestellt. Diese Entscheidung führte zu größeren Umstellungen bei Studenten und Lehrenden.

Die meisten Vorlesungen wurden auf Livestreaming umgestellt, während Veranstaltungen mit verpflichtender Präsenz großteils mittels Videokonferenzen abgehalten wurden. Eines der größten verbleibenden Probleme waren allerdings die schriftlichen Klausuren.

Im ersten Semester der Pandemie wurde dafür die Lernplattform Moodle verwendet, die bereits bei vielen Veranstaltungen in Verwendung war. Zusätzlich wurde von Studenten verlangt, eine Kamera hinter sich aufzubauen und sich bei der Durchführung von Klausuren zu filmen.

Dies führte allerdings schnell zu verschiedenen Unannehmlichkeiten, wie zum Beispiel unterschiedliche Konfigurationen für jede Klausur, eine systembedingte maximale Anzahl der Teilnehmer und organisatorische Schwierigkeiten in der Übersicht der Aufnahmen.

Um diese Probleme zu lösen haben wir am Institut für Systemsoftware eine Applikation namens "Xaminer" entwickelt. Das Programm wurde bewusst einfach gehalten, um kleinere Änderungen der internen Abläufe zu erleichtern und die Bedienung durch Laien zu ermöglichen.

1.1 Bestehende Systeme

Bevor wie die Details zu Xaminer erläutern werden, wir uns kurz mit den bereits existierenden Lösungen auseinandersetzen. Dabei werden wir uns hauptsächlich mit Moodle befassen, da dies an der JKU die meiste Verwendung findet. Es gibt weitere kommerzielle Systeme, auf die wir aber nicht genauer eingehen werden.

Frage 2

Unvollständig

Nicht bewertet

Frage markieren

For a finite automaton A , let

$C(A)$ denote the complement-automaton of A ,

$P(A)$ the power-automaton of A and

$L(A)$ the language accepted by A .

Which of the following statements are true?

Wählen Sie eine oder mehrere Antworten:

- 1. $L(C(P(A))) = \Sigma^* \setminus L(A)$
- 2. If A is deterministic, then $L(C(A)) = \Sigma^* \setminus L(A)$.
- 3. $L(P(C(A))) = \Sigma^* \setminus L(A)$
- 4. If A is complete, then $L(C(A)) = \Sigma^* \setminus L(A)$
- 5. For any A : $L(C(A)) = \Sigma^* \setminus L(A)$

Prüfen

Vorherige Seite
Nächste Seite

Abbildung 1.1: Eine Moodle Klausurfrage

1.1.1 Moodle

Moodle ist eine Open-Source-Lernplattform welche an der JKU intensiv wird. Die Plattform bietet eine große Auswahl an verschiedenen Aktivitäten, welche Studenten bei ihrer Weiterbildung unterstützen. Vor 2020 wurde Moodle an der JKU hauptsächlich verwendet, um Lernunterlagen und Skripts zu veröffentlichen und die Online-Abgabe von Übungen zu ermöglichen.

Da das System auch das Erstellen und Durchführen von Klausuren unterstützt (Abbildung 1.1), wurden viele schriftliche Klausuren so durchgeführt. Dies hat allerdings schnell zu Problemen geführt: Da Moodle etliche Konfigurationsmöglichkeiten bietet, haben Lehrende unterschiedlichste Einstellungen verwendet. Dies hat bei den Studenten zu Ungewissheiten geführt, welche Aktionen bei Klausuren erlaubt sind und welche nicht. So ist es bei den meisten - aber nicht allen - Klausuren möglich bereits beantwortete Antworten vor der Abgabe zu ändern.

Auch hat die kurzfristige Umstellung einige technische Mängel aufgezeigt. Wurde eine Klausur mit besonders vielen Teilnehmern durchgeführt, oder starteten mehrere Klausuren gleichzeitig, so war die gesamte Plattform für einige Minuten nicht zu erreichen. Dies wurde in folgenden Semestern verbessert indem die Infrastruktur ausgebessert wurde und alle Klausurteilnehmer eine leicht andere Startzeit bekamen.

2 Anforderungen

In der ersten Phase der Entwicklung von Xaminer haben wir diverse Anforderungen erhoben, um eine Übersicht über die Funktionalität und die Reichweite des Projekts zu erhalten. In diesem Kapitel erläutern wir für jede Benutzeroberfläche die einzelnen Bestandteile.

Das gesamte Projekt soll für alle Studenten und Lehrenden der Universität leicht zugänglich sein. Deswegen wurde entschieden, dass es sich bei Xaminer um eine Web-Applikation handeln soll.

2.1 Anmeldung

Da das gesamte Projekt sich mit Klausuren befasst, ist es wichtig, dass sämtliche Benutzer der Plattform eindeutig identifiziert werden können und nur auf die für sie bestimmten Inhalte und Funktionen Zugriff haben. Es soll kein Student eine Klausur sehen können, welche noch nicht begonnen haben. Auch sollen Lehrende keine Klausuren anderer sehen können.

Um dies sicherzustellen, könnte das Anmeldesystem der Universität, welches mittels den Technologien SAML oder LDAP funktioniert, verwendet werden. Wir haben uns allerdings entschieden, stattdessen eine simple selbst implementierte Anmeldemaske umzusetzen. Die Integration mit dem System der Universität würde lediglich das Erstellen eines neuen Accounts ersparen, wäre aber deutlich komplexer einzubinden.

Grundsätzlich wäre es kein Problem, wenn Unbekannte Personen sich als Lehrende ausgeben und eine Klausur erstellen, da diese nicht auf die Klausuren anderer Lehrenden Zugriff haben. Wir haben uns aber trotzdem entschlossen die Benutzer auf Bedienstete der Universität einzuschränken.

Die Anmeldung als Lehrender soll durch eine Email-Adresse der Universität und ein Passwort möglich sein. Hat ein Lehrender noch kein Passwort oder hat es vergessen, kann er es zurücksetzen, worauf eine Nachricht an die gegebene Email-Adresse gesandt wird. In dieser Nachricht kann die Änderung des Passworts bestätigt werden.

2.2 Übersicht

Angemeldete Lehrende sollen Zugriff auf eine Übersicht ihrer Klausuren haben. Für jede Klausur werden der Titel und das Durchführungsdatum angezeigt.

Es soll möglich sein, Klausuren anzulegen, zu bearbeiten, deren Ergebnisse herunterzuladen oder zu entfernen. Hierbei können nur zukünftige Klausuren bearbeitet werden und nur bei vergangenen Klausuren die Ergebnisse angezeigt werden.

2.2.1 Herunterladen und Entfernen

Bei jeder abgeschlossenen Klausur soll es möglich sein, die Ergebnisse in Textform herunterzuladen. Wählt ein Benutzer diese Option, sollen alle Ergebnisse in Textdateien und diese wiederum in ein ZIP-Archiv verpackt werden. Die automatische Auswertung dieser Daten würde den Rahmen dieses Projekts allerdings überschreiten und wird daher für die erste Version nicht umgesetzt. Hier gibt es viel Potential für eine Nachfolgearbeit.

Das Entfernen von Klausuren soll zu jeder Zeit möglich sein. Da dies allerdings immer mit Datenverlust verbunden ist, soll vor der Löschung eine Warnung angezeigt werden, bei der der Benutzer die Möglichkeit hat den Vorgang abubrechen. Bestätigt der Benutzer, dass die Klausur tatsächlich gelöscht werden soll, werden auch alle damit verbundenen Daten wie Teilnehmerlisten oder Ergebnisse gelöscht.

2.3 Bearbeiten

Ein Lehrender soll eine neue Klausur anlegen und ein Ersteller einer Klausur soll diese bearbeiten können. Beim Anlegen soll die Klausur Standardinhalten befüllt werden. Dabei wird der Beginn der neuen Klausur standardmäßig auf den nächsten Tag eingestellt.

Der Benutzer soll den Namen und das Durchführungsdatum der Klausur bearbeiten können. Anschließend können verschiedene Klausurelemente hinzugefügt und konfiguriert werden.

2.3.1 Elemente einer Klausur

Alle Elemente sollen mittels einfacher Textfelder bearbeitet werden können. Weiters soll rechts neben der Eingabe eine Vorschau angezeigt werden, welche das Element aus Sicht eines Studenten zeigt. Da unformatierter Text bei vielen Fragen unübersichtlich ist und die Einbindung von Bildern erschwert, haben wir uns entschieden, eine Technologie namens Markdown zu verwenden. Diese erlaubt es, mittels simpler Texteingaben häufig verwendete Elemente wie Überschriften, Bilder und Weblinks zu erstellen.

Xaminer soll die folgenden Klausurselemente anbieten:

- *Überschriften*, welche aus einem Textfeld bestehen und keine Eingabe von Studenten erwarten. Dieses Element ist aber nicht auf Überschriften beschränkt. Es können darin auch andere Inhalte, wie etwa ausführlichere Beschreibungen angegeben werden.
- *Textblock* besteht aus einer Angabe und einem Textfeld in dem Studenten eine Antwort eingeben können.
- *Single choice* Elemente erlauben es Studenten, aus einer der vordefinierten Antworten auszuwählen.
- *Multiple choice* Elemente erlauben es Studenten, aus einer oder mehreren der vordefinierten Antworten auszuwählen.
- *Image answer* ist ein Element, das als Nachfolgeprojekt zu dieser Arbeit im Projekt "A Shape-based Image Editor" von Michael Genser entwickelt wurde. Das Element erlaubt das Auswählen und Zuordnen von Bildern.

Bei allen Elementen außer Überschriften soll ein Lehrender auch Punkte angegeben können. Diese werden dem Studenten angezeigt, automatisch aufsummiert und am Ende der Klausur angezeigt.

Da wir nicht erwarten, dass alle Benutzer mit Markdown vertraut sind, soll auf Knopfdruck eine Liste an Markdown-Elementen inklusive Beispielen angezeigt werden. Wir sind der Überzeugung, dass dies, zusammen mit der Vorschau, Laien erlaubt, Klausuren ohne größere Probleme zu erstellen.

2.3.2 Status und Vorschau

Damit ein Benutzer sehen kann, wann sich die Klausur zuletzt geändert hat, soll eine Liste angezeigt werden, welche die letzten Status-Nachrichten anzeigt.

Weiters soll ein Benutzer die Möglichkeit besitzen, sich die gesamte Klausur aus der Sicht eines Studenten anzusehen. Dies soll auch vor dem angegebenen Klausurdatum funktionieren und mit anderen Personen geteilt werden können, ohne dass diese sich anmelden müssen.

2.4 Teilnehmer verwalten

Nachdem eine Klausur angelegt wurde, soll der Ersteller in einem sich öffnenden Fenster die Teilnehmer an der Klausur verwalten können. Dazu muss zuerst die Anzahl der Aufsichtspersonen festgelegt werden, damit die Teilnehmer gleichmäßig aufgeteilt werden können. Für jede Aufsichtsperson soll eine Email-Adresse und einen Video-Link (z.B. Zoom) zur Überwachung festgelegt werden. Diese Informationen werden später für das Informieren der Studenten verwendet.

2.4.1 Gruppen

Jede Aufsichtsperson betreut eine Gruppe von Studenten. Für jeden Studenten sollen folgende grundlegende Daten angezeigt werden:

- Matrikelnummer
- Studienkennzahl
- Vollständiger Name

- Email-Adresse

Ein Lehrender soll über Xaminer eine Email an einen ausgewählten Studenten senden können.

2.4.2 Import der Teilnehmer

Für den Import der Teilnehmerliste soll eine Datei im Comma-separated values Format ("CSV-Format") eingelesen werden. Dabei wird zunächst geprüft, ob die gelesene Datei die richtigen Spalten enthält, und falls dies nicht zutrifft wird eine Warnung ausgegeben. Dieses Format wurde gewählt, da das Kepler University Study Support System (KUSSS), welches an der Johannes Kepler Universität für Prüfungsanmeldungen verwendet wird, dieses Format unterstützt und dadurch sämtliche Teilnehmer sofort importiert werden können. Damit die Spalten in der CSV Datei den Werten zugeordnet werden muss die Datei Spalteüberschriften beinhalten und diese genau den folgenden Namen entsprechen:

- Matrikelnummer or MatNr
- Vorname
- Nachname
- E-Mail or Email
- SKZ

Damit das erstellen der CSV-Datei etwas erleichtert wird, soll Groß-Kleinschreibung bei Spalteüberschriften ignoriert werden.

Da es einen Anmeldeschluss im KUSSS gibt, kann davon ausgegangen werden, dass sich diese Teilnehmerlisten nicht mehr ändern. Es ist beabsichtigt, dass die Teilnehmer nur ein einziges Mal importiert werden, die Aufsichtspersonen allerdings danach noch einzelne Nachzügler nachtragen können.

Bei einem Import werden alle existierende Teilnehmer gelöscht und die neuen Studenten automatisch auf die Gruppen aufgeteilt.

Manuelles Hinzufügen von Teilnehmern

Zu jeder Gruppe können Studenten manuell hinzugefügt werden. Dies kann entweder einzeln, über dafür vorgesehene Eingabefelder geschehen, oder über einen Datei-Import. Dabei werden werden folgende Daten erfasst:

- Vorname
- Nachname
- Matrikelnummer
- Kennzahl der Studienrichtung
- Email-Adresse des Studenten

Die Email-Adresse kann verwendet werden, um den Studenten über die Klausur zu Informieren.

2.4.3 Emails

In der Email-Textbox wird ein vorgefertigter, aber editierbarer Text angezeigt, der als Email an ausgewählte Studenten oder an die Aufsichtspersonen gesendet werden kann. Der Text kann folgende Platzhalter enthalten, die von Xaminer mit den Daten des Adressaten bzw. der Klausur gefüllt werden:

- `#FIRSTNAME` Fügt den Vornamen des Studenten ein.
- `#LASTNAME` Fügt den Nachnamen des Studenten ein.
- `#EXAMNAME` Fügt den Namen der Klausur ein.
- `#STARTDATE` Fügt das Anfangsdatum der Klausur ein.
- `#STARTTIME` Fügt die Anfangszeit der Klausur ein.
- `#ENDTIME` Fügt das Enddatum der Klausur ein.
- `#STREAMLINK` Fügt die Endzeit der Klausur ein.

- *#EXAMLINK* Fügt den personalisierten Klausurlink für den Studenten ein. Dieser Link enthält einen Code der den Studenten eindeutig identifiziert und schwer zu erraten oder fälschen ist.

2.4.4 Status

Analog zur Klausurbearbeitung wird auch in der Teilnehmerverwaltung ein Statusfenster angezeigt. Hier wird auch bei jeder versandten Email eine Nachricht hinzugefügt.

2.5 Übersicht für Aufsichtspersonen

Die letzte Benutzeroberfläche, welche in der ersten Version von Xaminer umgesetzt wurde, ist eine Übersicht, die jeder Aufsichtsperson ihre teilnehmenden Studenten anzeigt. Die Aufsichtspersonen erhalten von Xaminer einen personalisierten Link, über den sie auf die Übersichtsseite gelangen.

2.5.1 Maßnahmen gegen Betrug

Um Betrug soweit wie möglich zu unterbinden, wurden in die Studentenansicht zwei Gegenmaßnahmen eingebaut.

Die erste ist eine markante Farbcodierung: Die Studierenden werden angewiesen, das Browser-Fenster zu maximieren, sodass man in der Videoüberwachung sehen kann, dass die linke Seite des Bildschirms dunkel und die rechte Seite hell ist. Dies wurde eingeführt, da wir bei vergangenen Online-Klausuren die Erfahrung gemacht haben, dass entweder die Kamera oder die Verbindung zum Studenten oft so schlecht ist, dass die Aufsichtsperson keine Details am Bildschirm lesen kann, sehr wohl aber das Hell-Dunkel-Muster erkennen kann.

Die zweite Maßnahme verhindert ein doppeltes Öffnen der Klausur. Dies wurde umgesetzt, indem die Klausurdaten nur einmal pro Studenten übertragen und anschließend lokal gespeichert werden. Das hat allerdings den Nachteil, dass die Teilnehmer bei einem Systemabsturz oder bei unbeabsichtigten Neuladen des Browserfensters nicht mehr in die Klausur einsteigen können. Deswegen erlaubt das Übersichtsfenster für Aufsichtspersonen, einen Teilnehmer wieder freizuschalten.

3 Datenstrukturen

Da Xaminer aus einer lokalen Browser-Oberfläche und einem Server besteht, mussten zur Kommunikation der beiden Teile Datenstrukturen definiert werden. Diese sollten sich möglichst an den Benutzerelementen orientieren und aus möglichst primitiven Datentypen bestehen, um zusätzliche Komplexität zu vermeiden.

3.1 Anmeldung

Für die Anmeldung werden lediglich die eingegebenen Daten ohne Änderung übertragen. Obwohl es sich hier um Passwörter handelt, sehen wir kein großes Problem, da die Verbindung zwischen Oberfläche und Service verschlüsselt ist.

3.2 Übersicht

Die Übersicht zeigt alle Klausuren des Benutzers an. Wenn wir diese komplett laden würden, würde dies zwar die Komplexität der Implementierung reduzieren, es könnte allerdings auch zu langen Ladezeiten und überflüssigen Datenübertragungen führen. Wir haben uns stattdessen dafür entschieden, immer nur jene Teile zu laden, welche auch tatsächlich angezeigt werden. Dies hat auch den Vorteil, dass die einzelnen Xaminer-Ansichten voneinander unabhängig sind und bei potentiellen Änderungen weniger Konflikte verursachen.

3.3 Bearbeiten

Der Bearbeitungsteil einer Klausur enthält die komplexesten Daten aller Eingabemasken. Das liegt daran, dass nicht nur eine große Anzahl verschiedener Texte übertragen werden, sondern auch daran, dass jede Art von Klausurelement eine andere Datenstruktur benötigt und deren Reihenfolge außerdem nicht bekannt ist. Dies haben wir dadurch gelöst, indem wir für jedes Prüfungselement zunächst dessen Typ senden, woraufhin die restlichen Werte je nach Bedarf gelesen werden.

3.4 Teilnehmer verwalten

Die Datenstruktur der Teilnehmer scheint auf ersten Blick aufgrund der vielen Einträge und verschiedenen Gruppen komplex. Im Gegensatz zu den Klausurelementen sind aber bei allen Teilnehmern die einzelnen Datenelemente bekannt.

3.5 Übersicht für Aufsichtspersonen

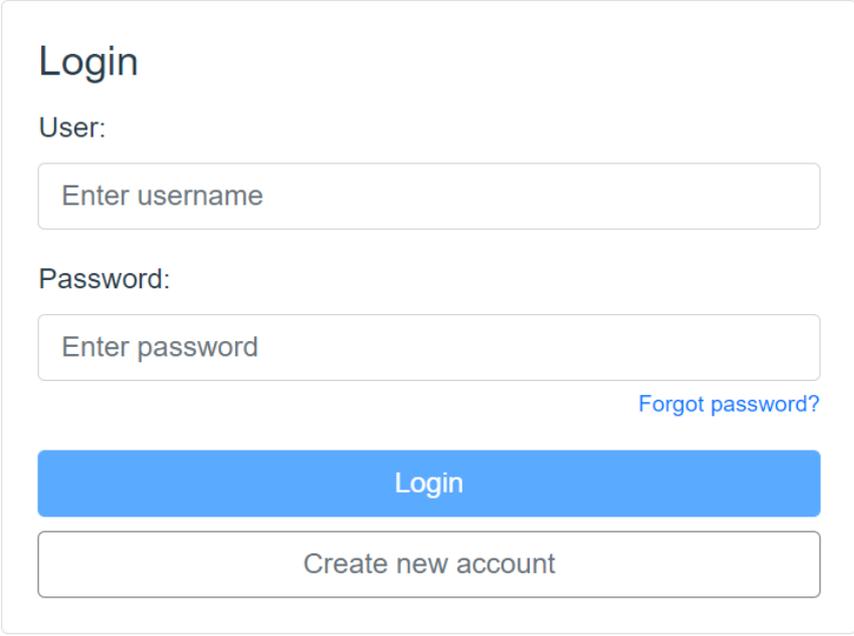
Bei der Übersicht für Aufsichtspersonen haben wir uns dazu entschieden, nur die zugewiesenen Studenten zu übertragen.

4 Design

Alle Benutzeroberflächen wurden mit Bootstrap entworfen. Bootstrap ist ein bekanntes Werkzeug zur Erstellung von Web-Applikationen. Es bietet eine Vielzahl an vorgefertigten Elementen und hilft dabei ein einheitliches Aussehen zu bewahren.

4.1 Anmeldung

Die Anmeldemaske, welche in Abbildung 4.1 zu sehen ist, wurde mit Absicht so gestaltet, dass sie der Anmeldung anderer Programme gleicht. Das soll den Benutzern helfen, sich schnell zurechtzufinden.



The image shows a login form with the following elements:

- Title:** Login
- User:** A text input field with the placeholder text "Enter username".
- Password:** A text input field with the placeholder text "Enter password".
- Forgot password?:** A blue link located to the right of the password field.
- Login:** A prominent blue button.
- Create new account:** A light gray button located below the login button.

Abbildung 4.1: Anmeldung

Upcoming Exams

Start	Name
10.01.2345 12:34	Future exam

Past Exams

Start	Name
19.11.2022 14:26	Past exam

- New
- Edit
- Manage groups and supervisors
- Download .zip with .txt submission files
- Download .zip with .html submission files
- Download .zip with .pdf submission files - This can take some minutes depending on participant count!
- Clone (Only content, not groups)
- Delete

Abbildung 4.2: Übersicht

4.2 Übersicht

Im Übersichtsfenster (Abbildung 4.2) ist links die Aufteilung von vergangenen und zukünftigen Klausuren zu sehen. Die Knöpfe auf der rechten Seite sind, bis auf den Knopf zum Erstellen von neuen Prüfungen, ausgegraut. Dies soll anzeigen, dass zuerst eine Klausur aus der Liste ausgewählt werden muss, bevor der Knopf gedrückt werden kann.

4.3 Bearbeiten

In Abbildung 4.3 ist die Eingabe der Grunddaten zu sehen. Die Funktionalität der einzelnen Elemente wird von Bootstrap bereitgestellt.

Abbildung 4.4 bis Abbildung 4.8 zeigen die einzelnen Klausurelemente. Jedes Element hat eine Eingabe auf der linken Seite und eine Vorschau, welche bei jeder Eingabe aktualisiert wird, auf der rechten Seite. Es sind die zu erreichenden Punkte des Klausurelements sowie die Knöpfe zum Verschieben und Entfernen einzelner Klausurelemente zu sehen.

Darunter befindet sich die Statusanzeige wie in Abbildung 4.9, sowie Knöpfe zum Anzeigen der Gesamtvorschau, zum Zurückkehren zur Übersicht und zum Speichern.

Exam name	Date	Start time	End time
<input type="text" value="Title"/>	<input type="text" value="20.11.2022"/> 📅	<input type="text" value="14:29"/> ⌚	<input type="text" value="14:29"/> ⌚

Abbildung 4.3: Die Eingabe der Grunddaten am oberen Ende der Oberfläche.

1. Text block	Points: <input type="text" value="0"/>	↑ ↓	Mock exam (x)
<input type="text" value="# Mock exam"/>			

Abbildung 4.4: Die Eingabefelder für eine Überschrift sowie die Live-Vorschau.

4.3.1 Vorschau der Gesamtklausur

In der Vorschau der Klausur (Abbildung 4.10) ist zu sehen, dass die linke Seite dunkel ist, während die rechte Seite hell ist. Wenn das Browserfenster maximiert ist, kann man dieses Muster leicht erkennen.

Am oberen Ende der Klausur befindet sich die Abgabe, welche die Eingabe des Namens erfordert. Dies wurde implementiert damit die versehentliche Abgabe unter falschem Namen unterbunden wird. Unter der Abgabe befindet sich noch der Name und Durchführungszeit der Klausur.

2. Text question Points:

Theory

What is the name of the following construct?
...

```
public string Name { get; set; }
```

...

1 Punkt | 1 Point

Theory

What is the name of the following construct?

```
public string Name { get; set; }
```

Abbildung 4.5: Eingabefelder für eine Textfrage

3. Code question Language: Points:

Classes

Implement a class named 'Point' with the fields 'x' and 'y'.

2 Punkte | 2 Points

Classes

Implement a class named `Point` with the fields `x` and `y`.

Abbildung 4.6: Eingabefelder für eine Codefrage

4. Single choice question Choices: Points:

Local variables

What is the result of the following code snippet?
...

```
Action<int> print;
for (int i = 0; i < 10; i++) {
  if (i == 0) {
    print = () => Console.WriteLine(i);
  }
}
print();
```

...

Compilation error

0

9

10

1 Mark correct answer

4 Punkte | 4 Points

Local variables

What is the result of the following code snippet?

```
Action<int> print;
for (int i = 0; i < 10; i++) {
  if (i == 0) {
    print = () => Console.WriteLine(i);
  }
}
print();
```

Choose one:

Compilation error

0

9

10

Abbildung 4.7: Eingabefelder für eine single choice Frage

5. Multiple choice question Choices: Points: ↑ ↓

Binary
What is contained in a compiled C# binary?

Native machine code

CIL instructions

MSIL instructions

Bytecode

1 Mark correct answer

4 Punkte | 4 Points

Binary

What is contained in a compiled C# binary?

Choose any:

Native machine code

CIL instructions

MSIL instructions

Bytecode

Abbildung 4.8: Eingabefelder für eine multiple choice Frage

Text block Add Markdown Help

Status

Overall points (calculated): 11 Full exam preview Go back to overview Save

Abbildung 4.9: Benutzeroberfläche unter den Klausursegmenten.

Submission:

Last Name + First Name Submit exam (final submit - can only be clicked once)

Exam: Future exam
10.01.2345, 12:34 - 23:45
MatNr: PreviewSubmission

Text question
Question

1

Abbildung 4.10: Vorschau

4.4 Teilnehmer verwalten

Die Benutzeroberfläche zur Verwaltung der Aufsichtspersonen und der damit verbundenen Gruppen ist in Abbildung 4.11 zu sehen. Wenn keine Email-Adresse oder kein Videokonferenz-Link eingetragen sind, werden die Eingabefelder gelb und mit Warnungssymbol markiert. Der blaue Kreis am linken Rand erlaubt es, eine Gruppe auszuwählen. Die Zahl daneben zeigt die Anzahl der zugewiesenen Studenten an.

Darunter, in Abbildung 4.12, befindet sich die Liste mit den Teilnehmern der Gruppe. Direkt daneben befinden sich die Eingabefelder zum Hinzufügen eines neuen Studenten.

Nachdem man Aufsichtspersonen und Teilnehmer konfiguriert hat kann man mittels der dafür vorgesehenen Textfeldern Informations-Mails an ausgewählte Personen schicken, wie in Abbildung 4.13 zu sehen.

	Email of supervisor	Zoom link	Admin link	Number of groups
<input checked="" type="radio"/> 13	<input type="text" value="roland.carlesi@jku.at"/>	<input type="text" value="https://jku.zoom.us/j/7458155"/>	▶ Group 1	<input type="text" value="4"/> <input type="button" value="Load Students from CSV"/> <i>i</i>
<input type="radio"/> 13	<input type="text" value="dacie.skirlin@jku.at"/>	<input type="text" value="https://jku.zoom.us/j/7458155"/>	▶ Group 2	
<input type="radio"/> 12	<input type="text" value="andrus.romanetti@jku.at"/>	<input type="text" value="https://jku.zoom.us/j/7458155"/>	▶ Group 3	
<input type="radio"/> 12	<input type="text" value="jessee.bagott@jku.at"/>	<input type="text" value="https://jku.zoom.us/j/7458155"/>	▶ Group 4	

Abbildung 4.11: Verwaltung der Aufsichtspersonen

Mat Nr	Skz	First Name	Last Name	Mail
82941531	521	Dan	Acosta	k82941531@students.jku.at
59461349	521	Libby	Askam	k59461349@students.jku.at
92824735	521	Dorolisa	Baggelly	k92824735@students.jku.at
21864634	521	Mar	Bediss	k21864634@students.jku.at
27400384	521	Grace	Blunsen	k27400384@students.jku.at
80964376	521	Eleanore	Bortolazzi	k80964376@students.jku.at
57053792	521	Letisha	Boshier	k57053792@students.jku.at
50371163	521	Auberta	Burton	k50371163@students.jku.at
21625336	521	Nonna	Cammimile	k21625336@students.jku.at
76911174	521	Waneta	Carlens	k76911174@students.jku.at
10099726	521	Bartholomew	Cauley	k10099726@students.jku.at
39631382	521	Vanny	Chatelot	k39631382@students.jku.at
56431087	521	Maia	Choppin	k56431087@students.jku.at

Delete Student
Add Student

Student ID

Study code

First name

Last name

Email

Abbildung 4.12: Verwaltung der Gruppen

Mail to students

Dear #FIRSTNAME #LASTNAME,

in the following you can find a Zoom-link as well as a personalized Exam-link for the exam '#EXAMNAME'.
DO NOT share this information with others!

Send to selected student
Send to students of all groups

Mail to supervisors

Dear supervisors,

you receive this information mail since you will supervise an exam group in the exam '#EXAMNAME'.

Send to selected supervisor
Send to all supervisors

Abbildung 4.13: Verwaltung der Email-Texte

4.5 Übersicht für Aufsichtspersonen

In der Maske in Abbildung 4.14 sieht man die Studenten der jeweiligen Gruppe. Es gibt keine Gruppenauswahl, da jede Aufsichtsperson nur die eigenen Teilnehmer sieht.

Für jeden Studenten ist zu sehen, ob dieser die Klausur begonnen und ob er sie abgeschlossen hat. Wenn die Klausur bereits begonnen wurde, kann der Zustand von der Aufsichtsperson zurückgesetzt werden, um einen Wiedereinstieg des Studenten zu ermöglichen.

First Name	Last Name	Mat Nr	Reactivate	Has Loaded Exam	Has Submitted
Dan	Acosta	82941531	<input type="button" value="Reactivate"/>	<input type="button" value="Not yet opened (since last server restart)"/>	
Libby	Askam	59461349	<input type="button" value="Reactivate"/>	<input type="button" value="Not yet opened (since last server restart)"/>	
Dorolisa	Baggelly	92824735	<input type="button" value="Reactivate"/>	<input type="button" value="Not yet opened (since last server restart)"/>	
Mar	Bediss	21864634	<input type="button" value="Reactivate"/>	<input type="button" value="Not yet opened (since last server restart)"/>	
Grace	Blunsen	27400384	<input type="button" value="Reactivate"/>	<input type="button" value="Not yet opened (since last server restart)"/>	
Eleanore	Bortolazzi	80964376	<input type="button" value="Reactivate"/>	<input type="button" value="Not yet opened (since last server restart)"/>	
Letisha	Boshier	57053792	<input type="button" value="Reactivate"/>	<input type="button" value="Not yet opened (since last server restart)"/>	
Auberta	Burton	50371163	<input type="button" value="Reactivate"/>	<input type="button" value="Not yet opened (since last server restart)"/>	
Nonna	Cammiemile	21625336	<input type="button" value="Reactivate"/>	<input type="button" value="Not yet opened (since last server restart)"/>	
Waneta	Carlens	76911174	<input type="button" value="Reactivate"/>	<input type="button" value="Not yet opened (since last server restart)"/>	
Bartholomew	Cauley	10099726	<input type="button" value="Reactivate"/>	<input type="button" value="Not yet opened (since last server restart)"/>	
Vanny	Chatelot	39631382	<input type="button" value="Reactivate"/>	<input type="button" value="Not yet opened (since last server restart)"/>	
Maia	Choppin	56431087	<input type="button" value="Reactivate"/>	<input type="button" value="Not yet opened (since last server restart)"/>	

Abbildung 4.14: Oberfläche für Aufsichtspersonen

5 Evaluierung

Um die Wirksamkeit von Xaminer zu Testen, wurde Xaminer für eine Klausur an dem Institut für Systemsoftware eingesetzt. Die Erstellung der Klausur sowie das Verwalten der Teilnehmer funktionierte dabei einwandfrei, es wurde aber deutlich, dass ein Klausur-element welches das Zuordnen von Begriffen und/oder Bildern hilfreich wäre. Bei der tatsächlichen Durchführung der Klausur hat sich gezeigt, dass die Zuordnung zu den Aufsichtspersonen gelungen ist.

Allerdings gab es einen technischen Zwischenfall der dazu geführt hätte, dass keine Abgaben gespeichert würden. Dieser Fehler wurde aber glücklicherweise noch während der Klausur entdeckt und behoben bevor die Teilnehmer gegangen sind. Es wurde sichergestellt, dass alle Abgaben korrekt gespeichert wurden und bei folgenden Klausuren ist kein vergleichbarer Fehler mehr aufgetreten.

Nach einigen Klausuren auch außerhalb des Institutes hat sich gezeigt, dass Xaminer ein effektives Werkzeug für die Durchführung von Online-Klausuren ist.