

**Gottfried Wilhelm
Leibniz Universität Hannover
Fakultät für Elektrotechnik und Informatik
Institut für Praktische Informatik
Fachgebiet Software Engineering**

Entwicklung einer interaktiven Echtzeit-Umfrageplattform für Dozierende

Bachelorarbeit

im Studiengang Informatik

von

Finn Mikoschek

**Prüfer: Prof. Dr. Schneider
Zweitprüfer: Dr. Jil Klünder
Betreuer: Alexander Specht**

Hannover, 16.09.2022

Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig und ohne fremde Hilfe verfasst und keine anderen als die in der Arbeit angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keinem anderen Prüfungsamt vorgelegen.

Hannover, den 16.09.2022

Finn Mikoschek

Zusammenfassung

Umfragen während den Vorlesungen können zum Verinnerlichen des Vorlesungsstoffes führen und bieten den Dozierenden ein direktes Feedback, inwieweit die Studierenden den Vorlesungsstoff verstanden haben. Umfrageplattformen wurden vor allem während des Online-Studiums aufgrund von Corona verwendet. Die Verwendung von Online-Umfragen soll nun auch im Präsenzunterricht mehr Anklang finden. Zwar bestehen schon Echtzeit Umfrage Plattformen, welchen es aber an bestimmten kostenfreien Features fehlt. Für dieses Problem soll eine uniinterne Lösung umgesetzt werden, welches bei der vorliegenden Arbeit erläutert wird. Ziel dieser Arbeit ist die Entwicklung einer interaktiven Echtzeit-Umfrageplattform für Dozierende. Um dieses Ziel zu erreichen, wurde eine Anforderungsanalyse durchgeführt welche eine Online-Umfrage beinhaltet die sich an Studierende und Dozierende gerichtet hat. Zudem wurden Use Cases zu den wichtigsten Szenarien der Applikation erstellt, um diese Kompakt und eindeutig darzustellen. Des weiteren wurde die Implementierung dokumentiert und im späteren Verlauf durch Testfälle evaluiert.

Abstract

Surveys during lectures can lead to internalisation of the lecture material and provide direct feedback to lecturers on the extent to which students have understood the lecture material. Survey platforms were mainly used during online study due to Corona. The use of online surveys should now also become more popular in face-to-face teaching. Although real-time survey platforms already exist, they lack certain free features. For this problem, a university-internal solution is to be implemented, which will be explained in this thesis. The aim of this thesis is to develop an interactive real-time survey platform for lecturers. To achieve this goal, a requirements analysis was carried out which included an online survey aimed at students and lecturers. In addition, use cases were created for the most important scenarios of the application in order to present them in a compact and clear way. Furthermore, the implementation was documented and later evaluated through test cases.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Problemstellung	1
1.3	Lösungsansatz	2
1.4	Struktur der Arbeit	2
2	Grundlagen	3
2.1	Django	3
2.2	React	3
2.3	Verwendete Bibliotheken	4
2.3.1	react-katex	4
2.3.2	jsdf	4
2.3.3	qrcode.react	4
2.3.4	Django-Rest-Knox	4
2.3.5	recharts	4
3	Anforderungsanalyse	5
3.1	Planung	5
3.2	Durchführung	6
3.3	Auswertung	6
3.4	UseCases	10
3.4.1	Use-Case zur Registrierung	10
3.4.2	Use-Case zum Fragen erstellen	12
3.4.3	Use-Case zum Umfrage erstellen	14
3.4.4	Use-Case zum veröffentlichen von Umfragen	15
3.4.5	Use-Case zum Exportieren der Umfrage Ergebnisse	16
4	Umsetzung	17
4.1	Implementierung	17
4.1.1	Registrierung und Login	17
4.1.2	Fragen bearbeiten und erstellen	18
4.1.3	Umfragen erstellen und bearbeiten	20
4.1.4	Umfragen veröffentlichen	22

5	Analyse	29
5.1	Test zum Use Case Registrierung	29
5.2	Test zum Use Case Frage erstellen	30
5.3	Test zum Use Case Umfrage erstellen	31
5.4	Test zum Use-Case Umfrage veröffentlichen	32
5.5	Test zum Use-Case Umfrageergebnisse Exportieren	33
6	Verwandte Arbeiten	35
7	Limitierung der Arbeit	37
8	Zusammenfassung und Ausblick	39
8.1	Zusammenfassung	39
8.2	Ausblick	39

Kapitel 1

Einleitung

Eine Echtzeitumfrage Plattformen bieten den Lehrenden eine Möglichkeit den Vorlesungsstoff durch interaktive Fragen an die Studenten zu vermitteln. Wodurch die lehrende Person, mit der Umfrage, Erkenntnis darüber erlangt, wie vertraut Studenten mit dem Vorlesungsstoff sind. Die Studenten erlangen des weiteren Kenntnis darüber, ob sie vertraut genug mit dem Vorlesungs-/Übungsstoff sind. Durch den Online-Unterricht und die damit verbundene Verwendung von Big Blue Button, konnten Umfragen leicht durchgeführt werden, da diese Funktion bereits bei dieser Plattform integriert war. Durch den aktuell wieder stattfindenden Präsenzunterricht wird die Plattform Big Blue Button nicht mehr verwendet, womit dieses Feature verloren ging. Die Entwicklung einer möglichen Lösung wird im folgenden dokumentiert.

1.1 Motivation

Vorlesungen vermitteln oftmals den Stoff linear. Wobei Studierende dem Gesagten zuhören und dies gegebenenfalls notieren. Dadurch dass der Vorlesungsstoff dabei oftmals mit Folien oder an der Tafel schriftlich dargestellt wird, um die Vermittlung bestimmtem Lehrstoff spannender und interaktiver zu gestalten, welches zudem auch noch anonym ist, wird ein Umfragetool benötigt, welches interaktiv in die Vorlesung eingebunden wird.

1.2 Problemstellung

Bisherige Echtzeit-Umfrageplattformen sind entweder nur begrenzt kostenfrei verfügbar, verfügen nicht über die benötigten Features, oder sind einfach nur unübersichtlich. Deshalb ist eine Entwicklung solch einer Plattform, die sich mit den wichtigsten Funktionen nicht hinter einer Paywall verstecken und grafisch gut dargestellt werden, sinnvoll.

1.3 Lösungsansatz

Um eine möglichst ansprechende Echtzeit Umfrage Plattform zu entwickeln, wurde zuerst eine Recherche zu einem passenden Web-Framework betrieben. Während dieser Zeit fand eine Anforderungsanalyse statt, um herauszufinden welche Features erwünscht werden und auf welchen Geräten/Browsern das Umfrage-Tool laufen sollte. Nach der Auswertung der Anforderungsanalyse wurde sich der Implementierung gewidmet, welche den Größten Teil dieser Arbeit ausmacht. Das erstellte Programm wurde zum Schluss anhand von Testfällen zu vorab erstellten Use-Cases getestet.

1.4 Struktur der Arbeit

In Kapitel 2 werden die Grundlagen der Arbeit beschrieben, 2. Die Grundlagen umfassen dabei das verwendete Framework, sowie die verwendeten Bibliotheken. In Kapitel 3 wird sich der Umfrage und dessen Auswertung gewidmet, welche die weitere Planung des Projektes, beispielsweise durch das Erstellen von Use-Cases, umfasst. Das 4 Kapitel geht anschließend auf die Implementierung des Projektes und mögliche Probleme ein. Kapitel 5 befasst sich mit der Testung des geschriebenen Codes. In Kapitel 6 geht es um die verwandte Arbeiten. In Kapitel 7 werden Grenzen sowie die Limitierung der Arbeit aufgezeigt. Das letzte Kapitel 8 fasst zum Schluss die Arbeit zusammen und bietet einen Ausblick in die Zukunft.

Kapitel 2

Grundlagen

In den Grundlagen wird das verwendete Framework und die verwendete Bibliotheken behandelt, welche zum Verständnis der Arbeit erforderlich sind.

2.1 Django

Das Django-Framework ist ein in Python geschriebenes Webframework[3], was dazu dient, die Entwicklung einer Website zu vereinfachen, indem zu integrierende Elemente von Django zu Verfügung gestellt werden. Das Django-Framework generiert ein Backend, in welchem dann sogenannte Apps erstellt werden können. Wird eine App erstellt generiert das genannte Framework automatisch verschiedene Dateien zu dieser Python. Die generierten Python-Dateien dienen zum einen der Erstellung von sogenannten Models, die dabei aus verschiedenen Feldern bestehen, welche bestimmte Datentypen annehmen können. Zu jedem dieser erstellten Models wird durch Django eine Datenbank-Tabelle in SQLite angelegt. Django bietet zudem eine grafische Benutzeroberfläche, um auf die Datenbanken zuzugreifen. In dieser Benutzeroberfläche können Objekte zu den Datenbanken hinzugefügt, bearbeitet und gelöscht werden. Des Weiteren verfügt Django über eine Nutzerverwaltung und die Verwaltung über deren Rechte. Das Framework wurde für die Entwicklung der interaktiven Echtzeit-Umfrageplattform als backend genutzt, um Daten zu verwalten.

2.2 React

React [4] ist eine JavaScript Programmbibliothek. Diese dient zur Erstellung von webbasierten Benutzeroberflächen. React stellt dabei verschiedene Bibliotheken zur Verfügung welche das Designen von Webseiten vereinfachen. React wurde bei dem Projekt als Frontend verwendet.

2.3 Verwendete Bibliotheken

Im Folgenden wird auf verwendete Bibliotheken eingegangen, welche bestimmte Funktionen bei der Implementierung vereinfacht haben.

2.3.1 react-katex

React-katex [7] ist eine Bibliothek welche zur Darstellung von Katex-Objekten in React dient. Katex ist die Haupt-Bibliothek, welche das Rendern von mathematischen Notation in Latex-Schreibweise ermöglicht. React-Katex stellt dabei vor generierte JSX Elemente zur Verfügung, welche in React integriert werden können, wodurch mathematische Notation grafisch dargestellt werden.

2.3.2 jsdf

JsPdf [2] ist eine javascript Bibliothek welche zur Generierung von PDF-Dateien dient. JsPdf-autotable [11] dient zudem dazu in die generierten Pdf-Dateien Tabellen hinzuzufügen. Die generierten Pdf-Dateien können dann im Browser dargestellt oder heruntergeladen werden.

2.3.3 qrcode.react

qrcode.react [9] ist ein React-Bestandteil, welcher zur Generierung von QR-Codes und dessen Darstellung dient. Dabei wird der genannte QR-Code über eine eingegebene URL generiert und leitet auf diese beim Scannen weiter. Zudem ermöglicht qrcode.react die grafische Anpassung des QR-codes, durch Veränderung der Größe und der Farbe.

2.3.4 Django-Rest-Knox

Das Django-Rest-Knox [8] dient zur Generierung von Nutzer-Tokens durch welche diese identifiziert werden können, um illegale Zugriffe zu verhindern. Die Nutzer- Tokens werden bei der Anmeldung von Nutzern generiert, wodurch diese dann eindeutig identifizierbar sind. Die Token dienen dabei als notwendiger Parameter, falls ein Datenzugriff auf Nutzer Interne Daten stattfindet.

2.3.5 recharts

Recharts [1] ist eine für React erstellte Bibliothek welche zur Darstellung von Graphen so wie Diagrammen verwendet wird.

Kapitel 3

Anforderungsanalyse

Die Anforderungsanalyse wurde anhand von einer Online-Umfrage sowie der Auswertung dieser durchgeführt. Des Weiteren wurden Use-Cases zu Grundfunktionen erstellt, um Programmabläufe kompakt darzustellen und diese später Testen zu können (siehe Kapitel 5). Im Folgenden wird auf die Planung des Projekte und der Durchführung sowie der Auswertung der Umfrage eingegangen. Zum Schluss wird die Erstellung von Use-Cases beschrieben.

3.1 Planung

Bei der Entwicklung eines Programmes ist die vorherige Planung essenziell, um im späteren Verlauf Zeit einzusparen. Der erste Schritt bei der Planung ist die Auswahl eines passenden Frameworks, die für Web-Applikationen geeignet ist und viele kompatible Bibliotheken zur Verfügung stellt. Aufgrund der großen Auswahl an Bibliotheken und der vorherigen Verwendung, wurde das Django-React Framework verwendet. Nach der Auswahl des passenden Frameworks, kommt es durch die Verwendung von Mock-ups zur Auswahl der essenziellen benötigten Funktionen. Hierbei wurden zum Verständnis der Funktionen die einzelnen Funktions-Szenarien grob skizziert. Zur genaueren Veranschaulichung der Szenarien und der einzelnen Möglichkeiten wurden außerdem Use-Cases erstellt, welche jeweils ein Szenario festhalten. Zudem dienen Use-Cases im weiteren Entwicklungsprozess als solide Vorlage für Testfälle, für welche diese dann auch genutzt wurden. Die Testung wird in Kapitel 5 behandelt. Um genauere Funktionen und Meinungen zu erhalten, wurde zusätzlich eine Online-Umfrage durchgeführt, welche im Folgenden behandelt wird.

3.2 Durchführung

Die Umfrage wurde über das Uni-interne Tool Survey-SE durchgeführt. Sie umfasst 14 verschiedene Fragen, die in vier verschiedene Kategorien unterteilt sind. Die erste Kategorie der Fragen beinhaltet persönliche Fragen. Daraufhin folgen allgemeine Fragen zur Nutzung von Umfrageplattformen, sowie der Verwendung von Geräten während der Vorlesung. Die dritte Kategorie umfasst Fragen zu anderen Umfrageplattformen und deren Nutzung. In dem letzten Abschnitt handeln die Abschlussfragen von bestimmten Features und weiteren Anmerkungen.

3.3 Auswertung

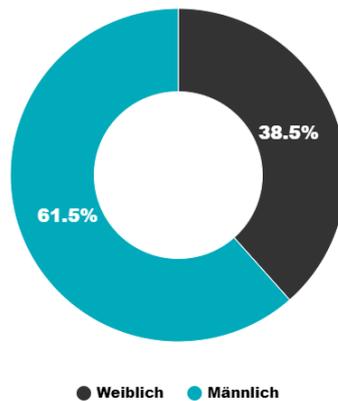


Abbildung 3.1: Verteilung des Geschlechtes der Teilnehmer

An der Umfrage nahmen 26 Personen teil, welche zum Zeitpunkt der Umfrage zwischen 18 und 31 Jahre alt waren. 24 von 26 Teilnehmern waren Studenten, eine Person war ein Schüler und die andere ein Dozierender. Die Mehrheit der Teilnehmer waren männliche Studenten, siehe Abbildung 3.1.

Die Befragten sollten anhand einer fünfstufigen Likert-Skala angeben, wie oft sie welches Gerät während der Vorlesung verwenden. Dies diente dazu die Priorität festzulegen für welche Geräte die Website als erstes ausgelegt und angepasst werden soll. Da diese sich bei verschiedene Formaten dementsprechend anpassen muss. Die Auswertung der Antworten wurde in einem Stacked-Balkendiagramm dargestellt (siehe Abbildung 3.2), in welchem die Stufen der Likert Skala Prozentual dargestellt sind. Durch die Auswertung wird deutlich, dass Handys am meisten während der Vorlesung zum Einsatz kommen. So gab lediglich nur eine Person an, dass sie nie ein Handy während der Vorlesung verwendet. 88% der Befragten verwenden mindestens gelegentlich ein Handy während der Vorlesung. Des weiteren ist bei der Abbildung der deutliche Nutzungsunterschied zwischen Tablet

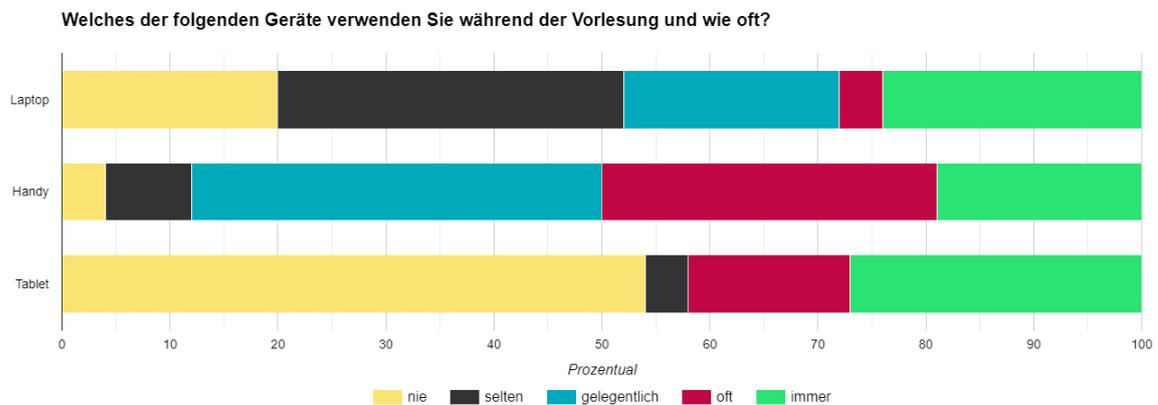


Abbildung 3.2: Geräte Nutzung während der Vorlesung

und Handy zu entnehmen. Bei diesem gaben 50% der Befragten an noch nie während der Vorlesung ein Tablet verwendet zu haben. Hierbei ist zu betrachten das Personen welche ein Tablet in der Vorlesung bereits verwendet haben dieses zu 58% bei jeder Vorlesung nutzen. Wird die Studie zum Besitz von Tablets, Laptops sowie Computern pro Haushalt in Deutschland [10] mit dem Besitz von Smartphones pro Person in Deutschland [12] verglichen, wird deutlich, dass der prozentuale Anteil bei Tablet-Besitzern deutlich geringer ist als der Anteil von Handy-Besitzern. Der Anteil von Handy-Besitzern liegt hierbei bei 88,8% und der Anteil von Tablet-Besitzern bei 25,6%. Der geringer Anteil von Tablet Besitzern könnte die 50% erklären welche kein Tablet während der Vorlesung verwenden. Laptops werden von über 50% der Teilnehmer nie oder selten verwendet. Dabei ist wie bei den Tablets wichtig zu betrachten das 47,27% der deutschen Haushalte einen Laptop besitzen [10]. Aus dem Balkendiagramm lässt sich schlussfolgern, dass die meisten Handys während der Vorlesung verwenden, da diese auch privat von dem Großteil der Bevölkerung benutzt werden. Des Weiteren wird ersichtlich, das die Teilnehmer am häufigsten ein Tablet bei jeder Vorlesung verwenden.

Die nächste Frage diente zur Ermittlung der Browser Verwendung der Teilnehmenden. Dabei wurde erneut eine fünfstufige Likert-Skala verwendet, welche ebenfalls anhand eines Balkendiagramms zur Veranschaulichung der Ergebnisse dargestellt wurde (siehe Abbildung 3.3). Diese Frage diente dazu, welche Browser bei der Implementierung von CSS Properties präferiert werden sollten. Da die Browser ab unterschiedlichen Versionen CSS Properties unterstützen und anders implementiert werden müssen. Auffällig ist hierbei das 79% der Teilnehmenden nie den Browser Edge und 88% nie den Browser Opeare verwenden. Aufgrund dessen wurden die beiden genannten Browser bei der Planung der grafischen Oberfläche vernachlässigt. Chrome, gefolgt von Safari und Firefox, wurde von den Teilnehmern am häufigsten verwendet

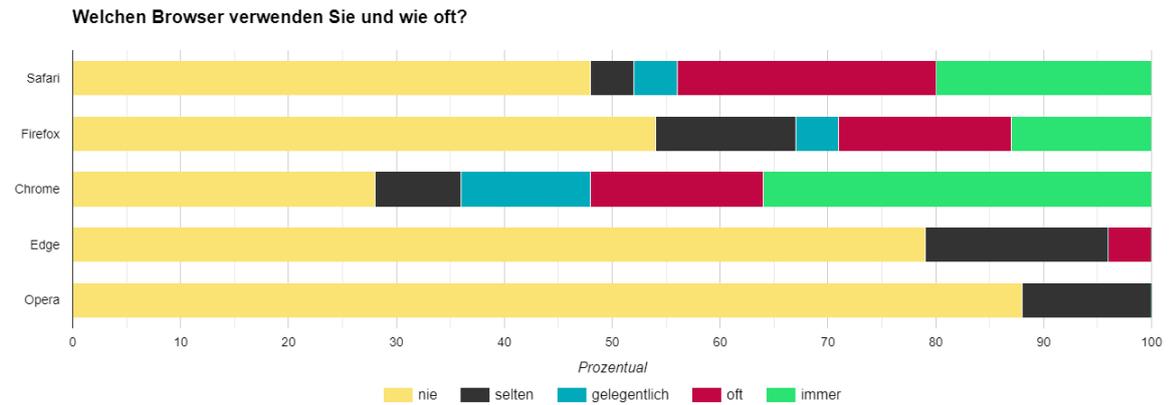


Abbildung 3.3: Browser Nutzung

und hat somit auch die höchste Prioritätsstufe bei der Planung der grafischen Benutzeroberfläche erhalten.

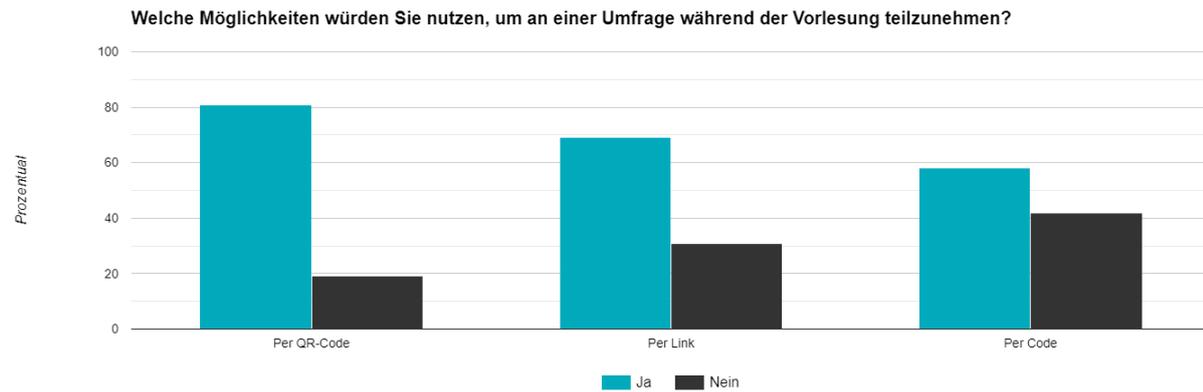


Abbildung 3.4: Teilnahme Möglichkeiten

Um weitere Priorität bei der Planung herauszufinden wurden die Teilnehmer anschließend gefragt welche Möglichkeiten sie nutzen würden um an einer Umfrage teilzunehmen. Die Ergebnisse wurden in einem mehrspaltigen Säulendiagramm dargestellt (siehe Abbildung 3.4). Am häufigsten gaben die Teilnehmer an, an einer Umfrage mit einem QR-Code teilzunehmen. Hierbei gaben 80% der Teilnehmer an, dass sie diese Möglichkeit nutzen würden. Ein Teilnehmer merkte zudem an, dass es vom Vorteil wäre jede Möglichkeit nutzen zu können. Bei der Planung der Implementierung wurde anhand der Auswertung der Frage die Teilnahme per QR-Code auf die höchste Prioritätsstufe gesetzt. Um die allgemeine Stimmung gegenüber Umfragen während Vorlesung und Übungen zu erfassen, wurde den Teilnehmer mit fünfstufiger Likert-Skala die Frage gestellt, wie sinnvoll sie Umfragen während Vorlesung und Übungen finden würden Abbildung 3.5. Dabei fanden die Teilnehmer

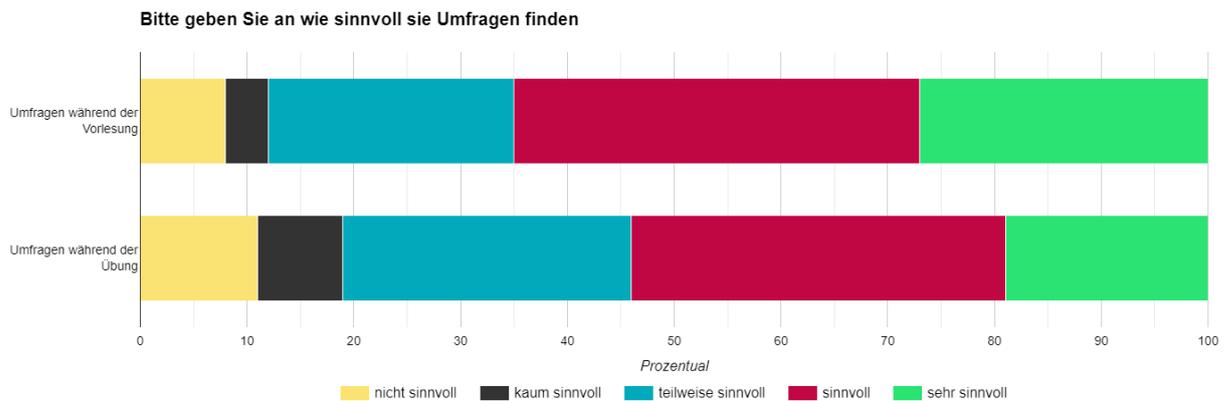


Abbildung 3.5: Wie sinnvoll sind Umfragen

Umfragen während der Vorlesung sinnvoller als während der Übungen. 12% fanden es gar nicht oder kaum sinnvoll während der Vorlesung Umfragen anzubieten und 18% waren bei Übungen dieser Meinung. Die Mehrheit gab an das Umfragen während der Übungen sowie der Vorlesungen sinnvoll oder sehr sinnvoll sind.

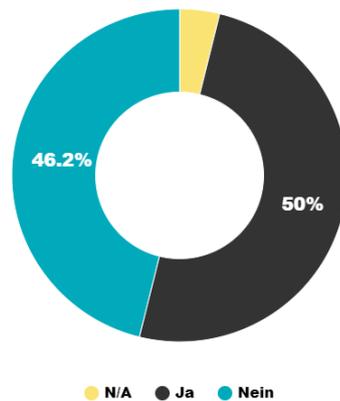


Abbildung 3.6: Verwendung einer Umfrageplattform während der Vorlesung

Des Weiteren wurden die Teilnehmer darüber befragt, ob sie schon eine Umfrageplattform verwenden haben. Und falls diese Aussage zutraf, wie oft sie diese verwendet haben, wenn sie die Möglichkeit dazu hatten, und ob ihnen das Umfrage Tool gefallen hat. Hierbei gab die Hälfte der Befragten an, dass sie bereits eine Umfrageplattform während der Vorlesung verwenden haben Abbildung 3.6. Die am häufigsten verwendete Plattform war hierbei Pingo welche 31% der Befragten, welche ein Umfrageplattform verwendet hatten, nannten. Weitere genannte Umfrageplattformen waren Evasys, ArSnova, Menti, Kahoot und Poll Everywhere. Allerdings wurden diese Umfrageplattformen jeweils nur einmal genannt und Pingo vier mal,

weshalb eine Auswertung der weiteren zwei Fragen zu den Umfrageplattformen aufgrund eines zu geringen Datensatzes unterlassen wurde.

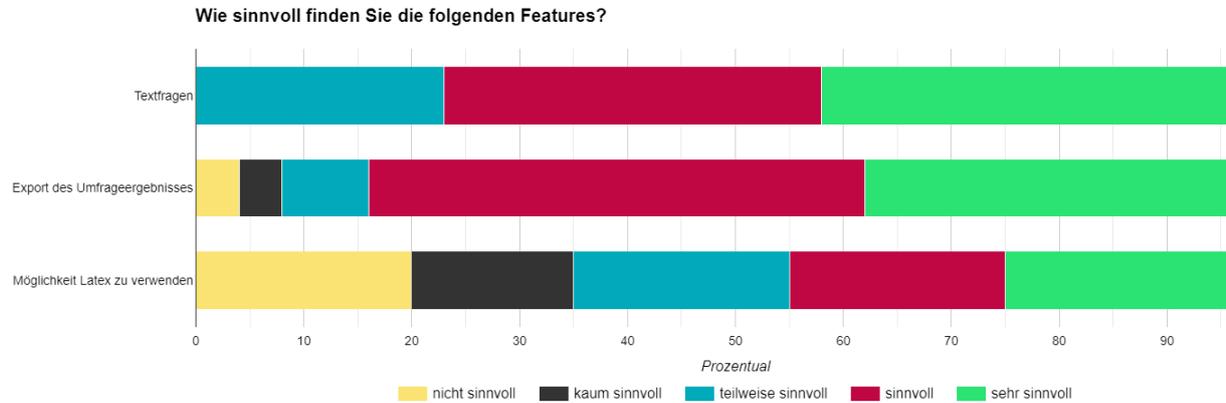


Abbildung 3.7: Sinnhaftigkeit genannter Features

Die nachfolgenden Fragen zielten darauf ab den Sinn bestimmter Features und die damit einhergehende Implementierung in die Website herauszufinden. Die Teilnehmer wurden befragt wie sinnvoll sie Textfragen, den Export von Umfrageergebnissen und die Möglichkeit LaTeX zu verwenden finden. Dabei konnten die Teilnehmer wieder mit fünfstufiger Likert-Skala bewerten, welche erneut als Stacked Balkendiagramm dargestellt wurden (siehe Abbildung 3.7). Über 80% der Befragten gaben an, dass ein Export der Umfrageergebnisse sinnvoll oder sehr sinnvoll ist. Ähnliches sah es bei dem Textfragen-Feature aus, bei welchem 77% der Meinung waren das es mindestens sinnvoll ist. Die Möglichkeit LaTeX zu verwenden fanden 45% sinnvoll oder sehr sinnvoll. Aufgrund dessen wurden die ersten beiden Features bei der Planung stärker priorisiert. Des Weiteren hatten die Teilnehmer noch die Möglichkeit eigene Feature-Vorschläge einzubringen. Z.B die Möglichkeit die Ergebnisse für Teilnehmer herunterzuladen oder die anonymisierte Teilnahme an der Umfrage. Welche Features implementiert wurden wird in Kapitel 4 behandelt.

3.4 UseCases

Um die benötigten Grundfunktionen des Programms zu veranschaulichen und diese besser im späteren Verlauf testen zu können, wurden Use-Cases erstellt.

3.4.1 Use-Case zur Registrierung

Der erste Use-Case befasst sich dabei mit der Registrierung eines neuen Nutzers und welche Szenarien dabei auftreten können. Eines der auftretenden Szenarien wäre z.B das Eingeben eines falschen Codes oder der Versuch

sich mit einer E-Mail zu registrieren, welche bereits registriert ist. Um diese Szenarien später besser Testen zu können wurden diese im Use-Case beschrieben.

Use Case Nr. 01	Registrierung
Erläuterungen	Nutzer möchte sich einen Account erstellen
Status	-
Systemgrenzen (Scope)	Gesamtsystem
Ebene	Hauptfunktion
Vorbedingung	Nutzer befindet sich auf der Registrierungsseite
Mindestgarantie	Dem Nutzer wird mitgeteilt, ob die Registrierung funktioniert hat
Erfolgsgarantie	Der Nutzer wird auf die Startseite weitergeleitet
Stakeholder	Nutzer
Hauptakteur	Nutzer
Auslöser	Der Nutzer gibt seine E-Mail ein
Hauptszenario	<ol style="list-style-type: none"> 1. Nutzer gibt seine E-Mail ein 2. drückt den Button "Bestätigen" 3. System sendet E-Mail mit Code an die angegebene E-Mail 4. System informiert Nutzer das ein Code an die angegebene E-Mail geschickt wurde 5. Nutzer gibt Code ein 6. Nutzer drückt den Button "Bestätigen" 7. System fordert den Nutzer auf, ein Passwort einzugeben 8. Nutzer gibt ein Passwort ein 9. Nutzer drückt den Button "Bestätigen" 10. System leitet Nutzer auf Startseite weiter
Erweiterung	<ol style="list-style-type: none"> 2a. Wenn keine E-Mail eingeben wurde, DANN <ol style="list-style-type: none"> 2a.1 System zeigt "Bitte geben Sie eine E-Mail an" 2b. Wenn die eingeben E-Mail nicht auf @inf.uni-hannover.de endet DANN <ol style="list-style-type: none"> 2b.1 System zeigt "E-Mail muss auf @inf.uni-hannover.de enden" 6a. Wenn der eingeben Code falsch ist, DANN <ol style="list-style-type: none"> 5a.1 System zeigt Nutzer an, dass der eingebende Code inkorrekt ist 9a. Wenn bereits ein Account mit der E-Mail besteht, DANN <ol style="list-style-type: none"> 9a.1 System zeigt Fehlermeldung
Priorität	unverzichtbar
Verwendungshäufigkeit	selten

3.4.2 Use-Case zum Fragen erstellen

Der zweite Use-Case Frage erstellen fasst das Fragen Erstellen in dem Umfrage-Tool mit bestimmten Rand Szenarien wie z.B dem nicht Ausfüllen des Fragen-Text-Feldes und wie sich das Programm dann verhalten soll, zusammen.

Use Case Nr. 02	Frage erstellen
Erläuterungen	Nutzer möchte eine Frage erstellen
Status	-
Systemgrenzen (Scope)	Gesamtsystem
Ebene	Hauptfunktion
Vorbedingung	Nutzer ist angemeldet
Mindestgarantie	Dem Nutzer wird mitgeteilt, ob die Erstellung der Frage funktioniert hat
Erfolgsgarantie	Frage wurde erstellt und wird bei dem Unterpunkt Fragen -bearbeiten angezeigt
Stakeholder	Nutzer
Hauptakteur	Nutzer
Auslöser	Der Nutzer klickt auf Fragen -erstellen
Hauptszenario	<ol style="list-style-type: none"> 1. Nutzer klickt auf Fragen -erstellen 2. Nutzer gibt einen Fragen-Text ein 3. Nutzer wählt aus, ob Frage in Latex dargestellt werden soll 4. Nutzer wählt Fragen-Typ aus 5. Nutzer gibt Antwort ein 6. Nutzer drückt den Button "Frage erstellen" 7. System erstellt Frage
Erweiterung	<ol style="list-style-type: none"> 4a. Wenn Single Choice oder Multiple Choice ausgewählt wurde, DANN <ol style="list-style-type: none"> 4a.1 System zeigt Antwort Objekt an, mit, Eingabefeld, Auswahlfeld für Latex und Auswahlfeld für korrekte Antwort. und einen Button zum Fragen hinzufügen 4b. Wenn Text Frage ausgewählt wurde, DANN <ol style="list-style-type: none"> 4b.1 System zeigt Antwort Eingabefeld an. 6a. Wenn der Fragen-Text leer ist, DANN <ol style="list-style-type: none"> 6a.1 System zeigt Fehlermeldung 6a.2 Nutzer gibt Fragen-Text ein 6a.2 Weiter bei 7 6b. Wenn Fragen Typ Single Choice und keine Antwort als korrekt ausgewählt wurde, DANN <ol style="list-style-type: none"> 6b.1 System zeigt Fehlermeldung 6b.2 Nutzer wählt korrekte Antwort aus 6b.2 Weiter bei 7
Priorität	unverzichtbar
Verwendungshäufigkeit	regelmäßig

3.4.3 Use-Case zum Umfrage erstellen

Der dritte Use-Case "Umfrage erstellen", deckt dabei das Verhalten und das Vorgehen beim Erstellen einer Umfrage ab. Mögliche Randszenarien sind das nicht Auswählen einer Frage und somit das nicht Fertigstellen der erstellten Umfrage.

Use Case Nr. 03	Umfrage erstellen
Erläuterungen	Nutzer möchte eine Umfrage erstellen
Status	-
Systemgrenzen (Scope)	Gesamtsystem
Ebene	Hauptfunktion
Vorbedingung	Nutzer ist angemeldet
Mindestgarantie	Dem Nutzer wird mitgeteilt, ob die Erstellung der Umfrage funktioniert hat
Erfolgsgarantie	Umfrage wurde erstellt und wird bei dem Unterpunkt Umfrage -bearbeiten angezeigt
Stakeholder	Nutzer
Hauptakteur	Nutzer
Auslöser	Der Nutzer klickt auf Umfragen -erstellen
Hauptszenario	<ol style="list-style-type: none"> 1. Der Nutzer klickt auf Umfragen -erstellen 2. Nutzer gibt einen Umfragen Titel ein 3. Nutzer wählt Fragen aus 4. System aktualisiert Fragen-Anzahl 5. Nutzer drückt den Button "Umfrage fertigstellen" 6. System erstellt Umfrage
Erweiterung	<ol style="list-style-type: none"> 5a. Wenn Umfragen-Titel unausgefüllt ist, DANN <ol style="list-style-type: none"> 5a.1 System zeigt Fehlermeldung an. 5a.2 Nutzer gibt Umfragen-Titel an 5a.3 Weiter bei 6 5b. Wenn Dauer unausgefüllt ist, DANN <ol style="list-style-type: none"> 5b.1 System zeigt Fehlermeldung an. 5b.2 Nutzer gibt Dauer an 5b.3 Weiter bei 6 5c. Wenn keine Frage ausgewählt wurde, DANN <ol style="list-style-type: none"> 5c.1 System zeigt Fehlermeldung 5c.2 Nutzer wählt Fragen aus 5c.3 Weiter bei 6
Priorität	unverzichtbar
Verwendungshäufigkeit	regelmäßig

3.4.4 Use-Case zum veröffentlichen von Umfragen

Der vierte Use-Case beschäftigt sich mit dem veröffentlichen und dem damit verbunden Vorbereiten der Umfrage. Zudem beschreibt der Use-Case knapp das beim Starten der Umfrage den Teilnehmern diese angezeigt wird.

Use Case Nr. 04	Umfrage veröffentlichen
Erläuterungen	Nutzer möchte eine schon erstellte Umfrage veröffentlichen
Status	-
Systemgrenzen (Scope)	Gesamtsystem
Ebene	Hauptfunktion
Vorbedingung	Nutzer ist angemeldet
Mindestgarantie	Umfragen werden angezeigt
Erfolgsgarantie	Umfrage wird für die Umfrageteilnehmer sichtbar
Stakeholder	Nutzer
Hauptakteur	Nutzer
Auslöser	Der Nutzer klickt auf Umfragen -veröffentlichen
Hauptzenario	<ol style="list-style-type: none"> 1. Nutzer klickt auf Umfragen -veröffentlichen 2. System zeigt erstellte Umfragen an 3. Nutzer klickt eine Umfrage an 4. System zeigt Umfragen Ansicht an 5. Nutzer klickt auf Umfrage vorbereiten 6. System zeigt Umfragen-vorbereiten Ansicht an 7. System setzt Umfrage auf public 8. Nutzer klickt auf Umfrage Starten 9. System startet die Umfrage 10. System zeigt Umfrage-Läuft Ansicht 11. System zeigt Umfrage für Teilnehmer an
Erweiterung	<ol style="list-style-type: none"> 2a. Wenn keine Umfrage vorhanden ist, DANN <ol style="list-style-type: none"> 2a.1 System zeigt keine Umfragen an.
Priorität	unverzichtbar
Verwendungshäufigkeit	regelmäßig

3.4.5 Use-Case zum Exportieren der Umfrage Ergebnisse

Der letzte Use-Case beschreibt das Beenden der Umfrage und das Auswerten der beendeten Umfrage sowie das Exportieren der Umfrageergebnisse.

Use Case Nr. 05	Umfrageergebnisse Exportieren
Erläuterungen	Nutzer möchte eine laufende Umfrage auswerten und die Ergebnisse Exportieren
Status	-
Systemgrenzen (Scope)	Gesamtsystem
Ebene	Nebenfunktion
Vorbedingung	Nutzer hat eine Umfrage gestartet
Mindestgarantie	Umfragen wird beendet
Erfolgsgarantie	Umfrageergebnisse werden als PDF exportiert
Stakeholder	Nutzer
Hauptakteur	Nutzer
Auslöser	Der Nutzer beendet die laufende Umfrage
Hauptscenario	<ol style="list-style-type: none"> 1. Nutzer klickt auf Umfragen Beenden 2. System beendet Umfrage 3. System zeigt, die Knöpfe Umfrage auswerten und erneut starten an 4. Nutzer klickt auf Umfrage auswerten 4. System zeigt Umfrage Statistiken an 5. Nutzer klickt auf exportieren 6. System generiert PDF 6. System öffnet Fenster zum speichern der PDF 7. Nutzer drückt auf speichern 8. System speichert PDF
Erweiterung	2a. Wenn Nutzer auf abbrechen drückt, DANN 2a.1 speichert das System das PDF nicht.
Priorität	unverzichtbar
Verwendungshäufigkeit	unregelmäßig

Kapitel 4

Umsetzung

4.1 Implementierung

Die Implementierung startet mit der Installation des Django-React Frameworks, welches sich in Frontend und Backend unterteilt. Das Backend befasst sich dabei mit der Datenspeicherung und der Verarbeitung. Das Frontend beschäftigt sich mit dem grafischen Darstellen dieser Daten, sowie dem Darstellen der Website.

4.1.1 Registrierung und Login

Dem Erstellen von Nutzeraccounts sowie das Anmelden mit den erstellten Benutzerdaten wurde sich zuerst gewidmet. Für die Nutzverwaltung wurde eine Django-interne Bibliothek verwendet, welche das Erstellen von Nutzern im backend erleichtert. Zur Authentifizierung der Nutzer wurde die Knox-Bibliothek verwendet, welche bei der Registrierung der Nutzer eindeutige Tokens für die Nutzer generiert. Damit diese dadurch identifiziert werden können. Des Weiteren gewährleistet die Verwendung der Tokens das Nutzer nur auf ihre eigenen Daten zugreifen können. Die Erstellung der Nutzer im Backend setzt sich somit aus der Verwendung der Django-internen User-Bibliothek und der Verwendung der Knox-Bibliothek zusammen. Die Registrierungsseite im frontend setzt sich aus mehreren Eingabefeldern und einem Submit Button zusammen. Beim Öffnen der Registrierungsseite ist nur das E-Mail-Eingabefeld, sowie der Bestätigungs-Button sichtbar. Beim Drücken des genannten Buttons wird im frontend überprüft ob eine E-Mail in das Eingabefeld eingegeben wurde und die E-Mail auf "@inf.uni-hannover.de"endet. Falls das Format nicht eingehalten wurde, erscheint ein Infotext, welcher den Nutzer darüber informiert das richtige Format zu verwenden. Wurde das Format eingehalten, sendet das Frontend die angegebene E-Mail an das Backend in welchem ein zufälliger sechsstelliger Code generiert wird. Das Backend sendet dann an die angegebene E-Mail eine Nachricht welche den zufälligen Code enthält. Das automatisierte Senden

der E-Mail wird dabei von einer Django-internen Bibliothek übernommen. Nach dem Senden der Mail wird der Code und die E-Mail Adresse in einer Datenbanktabelle gespeichert. Im Frontend wird das E-Mail-Eingabefeld zum Code-Eingabefeld umfunktioniert und es wird ein Infotext dargestellt, welcher den Nutzer darüber informiert das ihm ein Code an seine E-Mail gesendet wurde. Beim erneuten Betätigen des Bestätigungs-Buttons fordert das frontend vom Backend den vorher generierten Code an, um diesen mit dem im Eingabefeld vorhandenen Code zu vergleichen. Als Schlüssel dient dabei die E-Mail Adresse. Stimmen der Code aus der Datenbank und der Nutzeneingabe überein, ändert sich das Eingabefeld zu einem Passwort-Eingabefeld. Anderenfalls wird ein Infotext angezeigt, welcher den Nutzer darüber informiert, dass der eingegebene Code falsch war. Beim erneuten Drücken des Bestätigungs-Buttons wird überprüft ob das Eingabefeld leer ist. Wenn dies der Fall sein sollte, erscheint ein Fehlertext. Wurde ein Passwort eingegeben so wird der Wert aus dem Eingabefeld gelesen und mit der vorher eingegebenen E-Mail an das Backend übermittelt. Das Passwort wird mittels eines 256sha-Verfahrens im Backend verschlüsselt. Aus der E-Mail werden durch ein Split-Verfahren Vorname sowie Nachname entnommen. Diese Daten werden mit E-Mail und Passwort in der Datenbank gespeichert. Zudem wird der oben beschriebene Token generiert und als Rückgabewert an das Frontend zurückgegeben. Dieser wird im lokalen Speicher abgelegt und kann aus diesem gelesen werden. Nach der Fertigstellung der Implementierung der Registrierung wurde das Login-System implementiert, welches im Frontend aus zwei Eingabefeldern für E-Mail und Passwort, sowie einen Login-Button besteht. Hierbei wird bei Betätigung des Login-Buttons überprüft, ob beide Felder ausgefüllt wurden. Ist das der Fall, werden die Daten vom Frontend an das Backend übergeben. Das Backend überprüft dann durch Zugriff auf die Datenbank, ob die angegebene E-Mail existiert und ob das eingegebene Passwort dem abgelegten verschlüsselten Passwort entspricht. Wenn dies der Fall ist, sendet das Backend eine Bestätigung an das Frontend, welche den Nutzer-Token für den Zugriff auf die Nutzerdaten enthält. Bei falscher Eingabe erhält das Frontend vom Backend die Fehlermeldung 'incorrect Ingredients' und macht eine Fehlermeldung im Frontend sichtbar.

4.1.2 Fragen bearbeiten und erstellen

Nach der Fertigstellung der Nutzer Strukturen wurden im Backend Funktionen zur Fragen Erstellung implementiert. Hierfür wurde ein Fragen-Model im Backend implementiert. Dieses setzt sich aus einem Titel, einem Fragen-Typen, einem JSON Antwortfeld sowie einem Verweis auf das Nutzer-Objekt des Erstellers zusammen. In dem JSON Feld werden dabei die einzelnen Antworten in dem Format `id`: 0, `correct`: true, `text`: , `latex`: true gespeichert. Um Zugriff auf Fragen-Objekte aus dem Backend über das

Frontend zu erhalten, muss zur Verifizierung der Nutzer-Token übergeben werden, ohne das kein Zugriff auf die Fragen-Modelle möglich ist. Zur

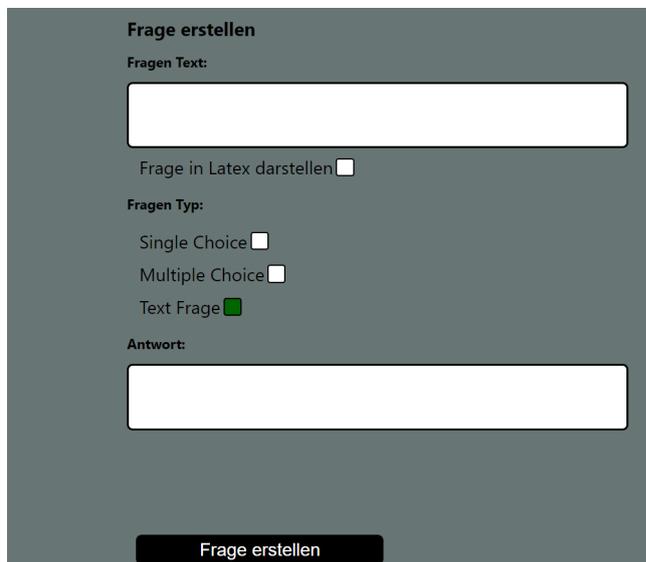


Abbildung 4.1: Text Frage erstellen

Erstellung und Bearbeitung der Frage wurden zwei Seiten im Frontend implementiert. Die erste dient zum Erstellen von Fragen und die zweite zum Bearbeiten von Fragen. Die Seite zum Erstellen von Fragen setzt sich aus einem Eingabefeld für die Frage, einem Auswahlfeld für den Typen der Frage sowie den Antwortfeldern zusammen (siehe 4.1). Durch Änderung des Fragen Typs werden im Frontend mehrere Methoden ausgeführt. Wird der Typ von Text-Frage auf Single oder Multiple Choice geändert, wird das Antwort-Eingabefeld entfernt und ein Antwort-Objekt wird erzeugt welches in Abbildung 4.2 dargestellt ist. Neben dem Antwort-Objekt wird ein weiterer Button sichtbar, welcher beim Aktivieren ein weiteres Antwort-Objektes generiert. Dieser Button ruft bei Betätigung eine Methode auf, welche das erste Antwort-Objekt über seine ID findet und dieses klonet. Nach der Klonung werden das Textfeld sowie die Checkboxes auf den Default Wert (empty) und false gesetzt. Des Weiteren wird ein interner Counter erhöht, welcher die Fragenanzahl symbolisiert. Der Aktuelle Fragen-Counterwert wird zudem als neue Id des geklonten Objektes festgelegt, um diesen identifizieren zu können. Das geklonte Objekt erhält ein weiteren Button durch welchen sich das Antwort-Objekt selbst löschen kann. Wird dieser Knopf gedrückt wird eine Methode aufgerufen, welche als Parameter den Knopf übergeben bekommt. Der übergebene Parameter wird zur Ermittlung der ID sowie der Löschung des Knopfes verwendet. Nach der Löschung des Knopfes werden alle nachfolgenden Antwortfelder bezüglich ihrer Name und ihrer ID geupdatet. Versucht der Nutzer eine Frage zu erstellen wird

Frage erstellen

Fragen Text:

Frage in Latex darstellen

Fragen Typ:

Single Choice

Multiple Choice

Text Frage

Antworten:

Antwort 1: Korrekte Antwort: Latex:

Antwort Hinzufügen

Frage erstellen

Abbildung 4.2: Single Choice Frage Erstellen

vorab überprüft ob ein Fragentext angegeben wurde. Ist dies nicht der Fall, wird der Nutzer darüber informiert das die Frage aufgrund von einem nicht vorhandenen Fragentext nicht erstellt werden konnte. Gleiches gilt für den Fall wenn der Nutzer eine Singelchoice Frage erstellen möchte und keine korrekte Antwort ausgewählt hat. Dabei wird der Nutzer darüber informiert das keine Frage erstellt werden konnte da keine korrekte Antwort ausgewählt wurde. Bei Multiplie Choice Fragen hingegen kann der Nutzer keine oder beliebig viele korrekte Antworten angeben. Versucht der Nutzer bei einer Single Choice Frage eine zweite Antwort als korrekt auszuwählen, so wird die vorherige korrekt ausgewählte Antwort als nicht korrekt geupdatet. Dies geschieht über eine Methode welche alle Antwortoptionen auf falsch zurücksetzt und dann die gedrückte Antwortoption auf wahr setzt. Werden alle Felder korrekt ausgefüllt, sendet das Frontend die erstellten Daten inklusive des Nutzer-Tokens an das Backend, in welchen dann ein Fragen-Objekt erstellt wird. Dieses wird anschließend in die Datenbank hinzugefügt. Das Bearbeiten von Fragen verhält sich im Backend so gut wie identisch zum Erstellen. Der einzige Unterschied liegt darin das keine neuen Fragen-Objekte erstellt, sondern bereits bestehende aktualisiert werden. Im Frontend wurde lediglich eine Suchleiste implementiert, über welche die erstellten Fragen gesucht und ausgewählt werden können. Wird eine Frage ausgewählt, wird diese mit den aktuellen Werten angezeigt. Dabei werden die Daten der Frage vom Backend an das Frontend übertragen.

4.1.3 Umfragen erstellen und bearbeiten

Die erstellten Fragen können dann in unterschiedliche Umfragen eingebettet werden. Umfragen können wie die Fragen erstellt, gelöscht und bearbeitet

werden. Die unterschiedlichen Funktionen können links in der Sidebar ausgewählt werden, wie in Abbildung 4.3 zu sehen ist. Das Umfragen-

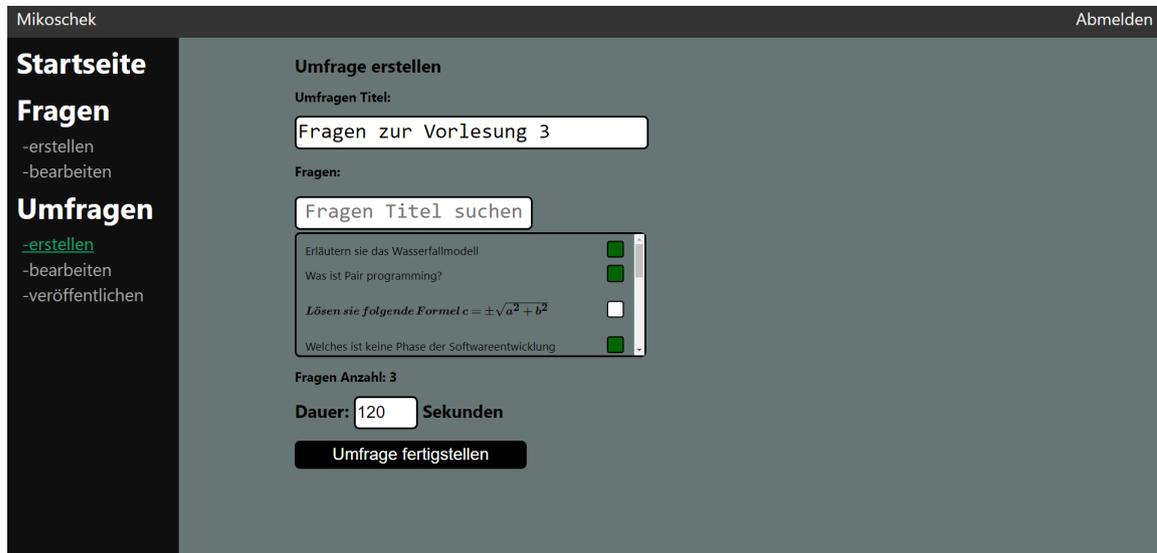


Abbildung 4.3: Umfragen erstellen

Modell besteht aus einem Umfragentitel, einem ManytoMany Field von Fragen sowie der Bearbeitungsdauer. Wie bei dem Fragen-Modell besitzt das Umfragen-Modell ebenfalls einen verweis auf den Umfragen-Ersteller. Wird die Seite zum Umfragen erstellen ausgewählt, werden dabei aus dem Backend die von dem angemeldeten Nutzer erstellten Fragen an das Frontend gesendet und über dieses im Fragenfeld dargestellt, wie in Abbildung 4.3 zu sehen ist. Zur Gewährleistung, dass auch nur Fragen angezeigt werden welche der Nutzer selbst erstellt hat, wird der Nutzer-Token mit an das Backend übergeben. Bevor die Fragen-Objekte im Frontend dargestellt werden, wird über den Parameter Latex überprüft, ob das Fragen-Objekt in Latex geschrieben wurde. Falls dies der Fall ist, wird es grafisch mit react-katex Bibliothek dargestellt. Jedes Fragen-Objekt wird dabei mit der ID der ausgewählten Frage versehen, was zur späteren Identifikation dient. Das Fragen-Objekt im Frontend setzt sich aus dem Fragentitel sowie einer Checkbox zusammen und wird in eine Liste eingefügt. Die Liste kann dann per Suchleiste durchsucht werden (siehe 4.3). Der Nutzer kann per Drücken der Checkbox der jeweiligen Frage die Frage zu der Umfrage hinzufügen. Beim Auswählen der Frage zählt ein eingebettet Counter die Fragenanzahl hoch. Wird der Knopf "Umfrage fertigstellen" gedrückt, wird eine Methode aufgerufen, welche überprüft bei welchen Fragen-Objekten die Checkbox angeklickt wurde. Von den ausgewählten Fragen-Objekten wird dabei die ID gelesen, durch welche das vorher abgespeicherte Fragen Objekt von der Datenbank identifiziert werden kann und in ein Array eingefügt wird. Das

genannte Array wird dann mit den restlichen Umfragewerten vom Frontend an das Backend übergeben. Bevor die Umfrage auf der Datenbank abgelegt wird, wird die eindeutige ID der Umfrage, welche durch Django festgelegt wurde, zusätzlich als Code der Umfrage festgelegt, über welchen Teilnehmer an der Umfrage Teilnehmen können, falls diese gestartet wurde. Zusätzlich wird beim Erstellen der Umfrage noch ein weiteres Objekt mit dem Namen Polltime im Backend erzeugt. Das Objekt speichert die Umfragenzeit und einen Wert ob die Umfrage gestartet wurde. Zudem hat das Objekt die identische ID zum Umfragen-Objekt. Das Objekt wird bei der Teilnahme an der Umfrage per QR-Code, Url oder Code gelesen und gibt die noch verfügbare Bearbeitungszeit zurück und ob die Umfrage bereits gestartet wurde. Das Umfragenbearbeiten verhält sich wie das Fragenbearbeiten. Es werden bloß Umfrage-Objekte anstatt Fragen-Objekte dargestellt.

4.1.4 Umfragen veröffentlichen

Beim Veröffentlichen der Umfrage durchläuft der Ersteller mehrere Szenarien. Wählt der Nutzer in der Sidebar den Unterpunkt Umfrage veröffentlichen aus, werden ihm alle von ihm erstellten Umfragen aufgelistet.

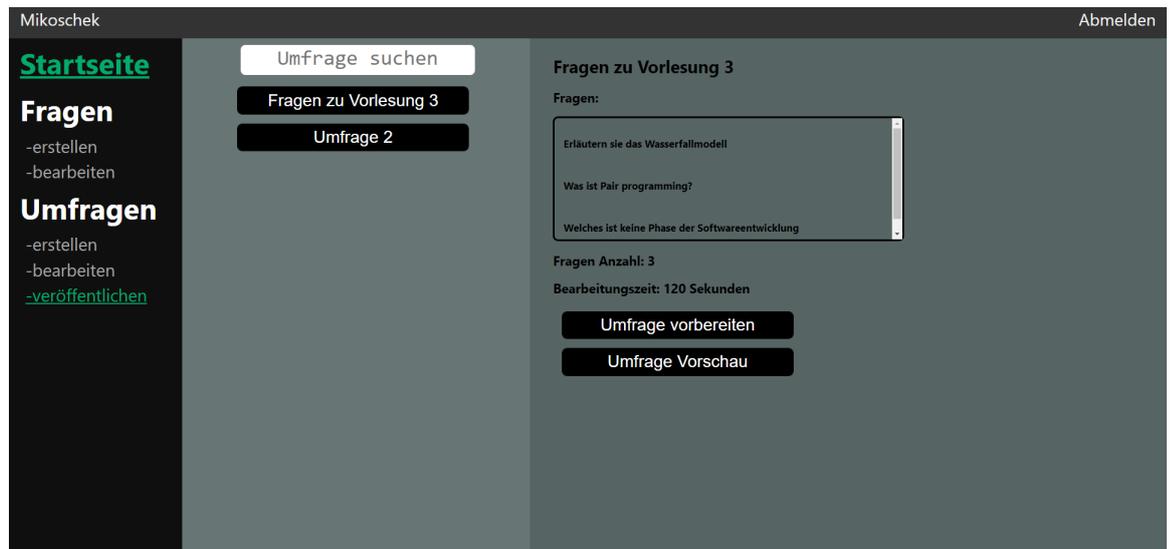


Abbildung 4.4: Umfragen veröffentlichen

Die Umfragen werden dabei als Knöpfe angezeigt, welche mit der Umfrage-ID identifizierbar sind. Die erstellten Umfragen können zudem in der Suchleiste gesucht werden. Wird eine Umfrage ausgewählt wird diese über die Knopf ID identifiziert und die Umfragedaten werden von der Datenbank aus dem Backend ins Frontend geladen und angezeigt. Rechts neben der Umfrageliste wird dabei die ausgewählte Umfrage angezeigt. Diese wird wie

folgt dargestellt 4.4.

Fragen zu Vorlesung 3

Frage 1: Erläutern sie das Wasserfallmodell

Frage 2: Was ist Pair programming?

Frage 3: Welches ist keine Phase der Softwareentwicklung

- Anforderungsphase
- Konzeptionsphase
- Entwicklungsphase
- Entspannungsphase

Umfrage beenden

Abbildung 4.5: Umfragen Vorschau

Wird die Taste "Umfragen vorschau" gedrückt bekommt der Nutzer in einem neuen Fenster die vom ihm erstellte Umfrage so angezeigt wie sie auch die Teilnehmenden sehen würden. Dadurch kann der Ersteller die Umfrage auf mögliche Fehler untersuchen, wie z.B. Latex Syntax Fehler, bevor diese veröffentlicht wird. Die Umfragenvorschau würde für die in Abbildung 4.4 dargestellten Umfrage, wie folgt aussehen 4.5. Der einzige Unterschied zu der tatsächlichen Umfrage liegt darin, dass die verbleibende Zeit nicht angezeigt wird, sowie das der Umfragebeenden-Knopf deaktiviert ist. Wird der andere Knopf "Umfrage vorbereiten" betätigt, wird der Nutzer zu einer anderen Sicht weitergeleitet. Bevor dies allerdings passiert, wird die ausgewählte Umfrage auf öffentlich gesetzt. Das bedeutet, dass die Teilnehmer die Umfrage nun über den Zugangscode, URL und QR-Code finden können. Dies passiert indem das Frontend das geupdatete Umfrage-Objekt an das Backend sendet und in diesem der Parameter `public` auf `true` gesetzt wird. Des Weiteren werden alle, falls vorhanden, bestehenden Umfrageergebnisse von der Umfrage aus der Datenbank gelöscht. Die Umfrageergebnisse welche eine eigene Datenbanktabelle besitzen, werden dabei über die Umfragen-ID identifiziert

und können so gelöscht werden.



Abbildung 4.6: Umfragen Vorbereitung

Die Umfragenvorbereitung dient dabei dafür, dass alle Studierende, welche teilnehmen wollen, die Umfrage über QR-Code, Link oder Code aufrufen können. Diese sehen die Umfrage allerdings erst nachdem diese gestartet wurde. Dies wird über das erwähnte Objekt `Politime` sichergestellt welches Umfragenzeit sowie einen Booleanwert enthält, welche aufzeigt, ob die Umfrage gestartet wurde. Der QR-Code wird über die `qrcode.react` Bibliothek dargestellt. Die Umfragenerstelleransicht sieht wie folgt aus 4.6. Der QR-Code kann in einem neuen Fenster angezeigt werden. Dies soll dazu dienen das bei laufender Umfrage die Teilnehmer auch weiterhin per QR-Code beitreten können. Der Teilnehmer Code und der Link werden in dem neuen Fenster zudem ebenfalls angezeigt. Wird der Knopf 'Umfrage Starten' gedrückt wird das zu der Objekt-`Politime` geändert und der parameter `started` wird dabei auf `true` gesetzt. Durch die Änderung dieses Parameter erscheint bei den Teilnehmer, welche schon auf der Umfragenseite warten, die Umfrage. Das Erscheinen der Umfrage ist dabei maximal um eine Sekunde zu der Veröffentlichung verzögert. Dies liegt daran, dass bei den wartenden Nutzern sekundlich ein für die Umfrage eindeutiges `Politime`-Objekt aus dem Backend ans Frontend gesendet wird, welchem entnommen wird, ob die Umfrage gestartet ist. Die sekundliche Abfrage der Backenddaten wird dabei über ein im Frontend laufendem Intervall umgesetzt, welches sekundliche eine Methode zum erhalten der Backenddaten ausführt. Die Teilnehmer haben nun eine Ansicht wie in 4.5 gezeigt. Allerdings ist diese wie beschrieben mit minimalen Unterschieden. Das bereits beschreibende `Politime`-Objekt wird nun sekundlich geupdatet. In dieses wird die verbleibende Zeit geschrieben. Die verbleibende Zeit wird dabei von dem Ersteller der Umfrage entnommen, bei welchem im Frontend ein ebenfalls ein Intervall läuft welches den Timer



Abbildung 4.7: Laufende Umfrage (Ersteller Sicht)

sekündlich um eins runter zählt und diesen Timer dann in das Polltime-Objekt schreibt. Das Polltime-Objekt wird dann wieder sekündlich von den Teilnehmer gelesen. Dies passiert auch über ein Intervall welches sekündlich die Daten von dem Polltime-Objekt aus dem Backend holt. Das garantiert das alle Teilnehmer die gleiche Restzeit haben. Das führt dazu, dass es eine Rolle spielt, wann ein Teilnehmer beigetreten ist. Anderenfalls wäre es kein Echtzeit Umfrage-Tool. Der Fragenersteller hat bei einer laufenden Umfrage eine Ansicht wie in Abbildung 4.7 dargestellt. Bei welcher zudem eine Übersicht darüber besteht wie viele Teilnehmer an den unterschiedlichen Fragen teilgenommen haben. Die Teilnahmewerte an der Umfrage werden wie folgt ausgewertet: Textantworten welche beim Beenden der Umfrage nicht ausgefüllt wurden, werden nicht übermittelt und somit auch nicht gespeichert. Alle Multiple Choice sowie Single Choice Antworten werden übermittelt. Dabei zählen bei einer Frage zwei angeklickte Antworten nur als eine Teilnahme. Alle nicht angeklickten Felder werden als falsch gewertet. Beendet der Teilnehmer die Umfrage, werden die oben beschriebenen Teilnehmerwerte an das Backend übermittelt und von dem Frontend von dem Umfragenersteller gelesen. . Des Weiteren werden auch in der Umfragesicht Link sowie Code zu der Umfrage eingeblendet. Falls der Ersteller der Umfrage die Live-Statistiken einblenden möchte und Teilnehmer weiterhin teilnehmen können. Die Umfrage kann durch zwei Möglichkeiten beendet werden. Die erste Möglichkeit ist das Auslaufen der Zeit. Dabei wird die Umfrage bei null Sekunden gestoppt. Dies wird wie folgt gewährleistet: der Timerwert des Erstellers wird intern über das genannte Intervall überprüft. Liegt der Wert bei null wird das Intervall des Erstellers beendet und das Polltime-

Objekt wird ein letztes mal aktualisiert. Dabei wird die verbleibende Zeit auf null gesetzt und der parameter started auf false. Das Intervall der Nutzer überprüft auch sekundlich ob die verbleibende Zeit über null liegt und ob der paramter started true ist. Ist das nicht der Fall, wird dieses Intervall ebenfalls beendet und die Umfrage ist für den Teilnehmer nicht mehr sichtbar. Fast identisch läuft es bei dem Beenden der Umfrage über den Beendenknopf ab. Hierbei wird lediglich die verbleibene Zeit auf null gesetzt, wodurch sich der Rest, wie oben beschrieben, verhält. Nach dem Beenden der Umfrage hat der Ersteller die Möglichkeit die Umfrage erneut durchzuführen oder sich die Umfragenstatistiken anzeigen zu lassen. Bei erneuten Starten wird die verbleibende Zeit durch die Umfragezeit ersetzt und auch so ins Polltime-Objekt geschrieben. Des Weiteren werden die voran gegangen Umfragestatistiken gelöscht. Wird der Umfragestatistiken-



Abbildung 4.8: Umfrage Statistiken

Knopf gedrückt werden alle abgegebenen Antworten zu der Umfrage aus dem Backend von der Datenbank gelesen. Diese werden dabei wieder über die Umfragen-ID identifiziert. Die Umfrage-Objekte werden dann für den Ersteller angezeigt 4.8. Textuelle Antworten werden dabei einzeln aufgelistet und Multiple Choice sowie Single Choice per Graph pro Frage dargestellt. Die Graphen werden dabei per recharts Bibliothek dargestellt.

Der Ersteller hat in der Statistiken Ansicht die Möglichkeit sich die korrekten Antworten anzeigen zu lassen. Dabei hat jede Frage einen dafür vorgesehenen Knopf, welcher für die ausgewählte Frage die Antwort präsentiert. Wird der Knopf betätigt, wird bei den textuellen Antwort der von dem Professor vorher verfasste Text grün und bei dem Balkendiagrammen werden die Balken ebenfalls grün, welche die korrekt Antwort darstellen.

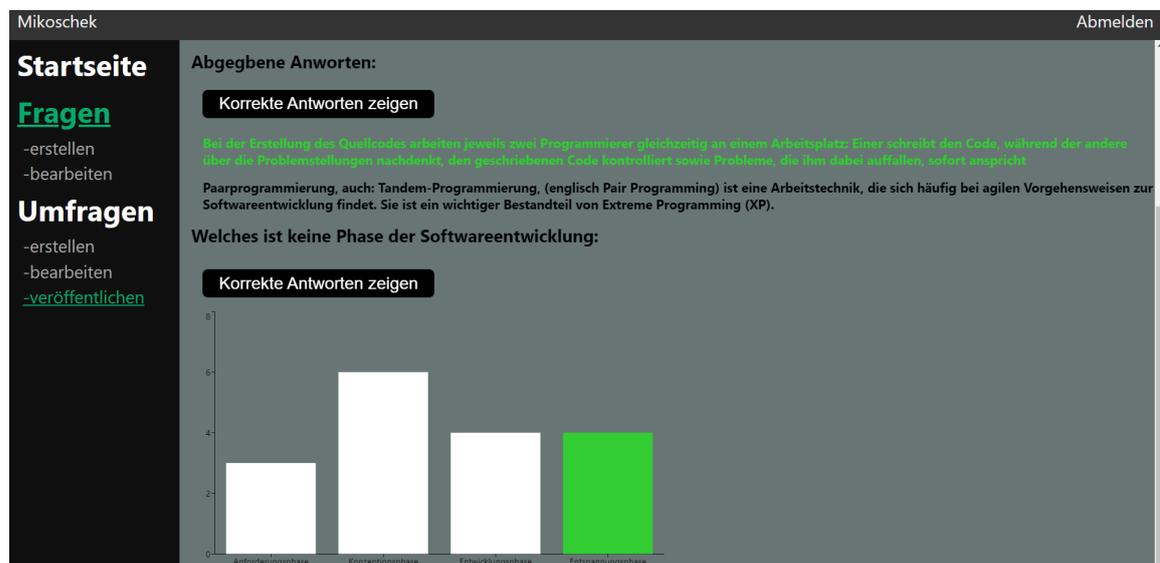


Abbildung 4.9: Umfrage Statistiken korrekte Antworten

Wie in Abbildung 4.9 dargestellt. Das nicht direkte Anzeigen der korrekten Ergebnisse bietet vor allem bei textuellen Antworten die Möglichkeit über diese zu diskutieren und gemeinsam die richtige Lösung rauszufinden.

Die gesammelten Ergebnisse können über den exportieren Button als Pdf heruntergeladen werden. Dies geschieht über die JSspdf und die jspdf-autotable Bibliothek. Hierbei werden für jede Frage Tabellen angelegt, welche die Teilnehmeranzahl und die Korrektheit der Antwort beinhalten. Die automatisch generierte PDF würde zum Beispiel wie in Abbildung 4.10 aussehen.

 Fragen zu Vorlesung 3

Erläutern sie das Wasserfallmodell:

Antworten	Korrekt
	true
Ein Wasserfallmodell ist ein lineares (nicht iteratives) Vorgehensmodell, das insbesondere für die Softwareentwicklung verwendet wird und das in aufeinander folgenden Projektphasen organisiert ist.	false

Was ist Pair programming?:

Antworten	Korrekt
Bei der Erstellung des Quellcodes arbeiten jeweils zwei Programmierer gleichzeitig an einem Arbeitsplatz: Einer schreibt den Code, während der andere über die Problemstellungen nachdenkt, den geschriebenen Code kontrolliert sowie Probleme, die ihm dabei auffallen, sofort anspricht	true
Paarprogrammierung, auch: Tandem-Programmierung, (englisch Pair Programming) ist eine Arbeitstechnik, die sich häufig bei agilen Vorgehensweisen zur Softwareentwicklung findet. Sie ist ein wichtiger Bestandteil von Extreme Programming (XP).	false

Welches ist keine Phase der Softwareentwicklung:

Antwort	Ausgewählt	Korrekt
Anforderungsphase	3	false
Konzeptionsphase	6	false
Entwicklungsphase	4	false
Entspannungsphase	4	true

Abbildung 4.10: Generierte PDF Datei zu der Umfrage

Kapitel 5

Analyse

Nach der Implementierung sowie während der Implementierung kam es zu ständigen Testungen der Software. Dies geschah zum Einen über klassische Print-Ausgaben welche den Rückgabewert von Methoden auf der Konsole ausgaben. Diese Ausgaben wurden dann mit den Sollwerten der Funktionen verglichen, um so die Korrektheit der Funktionen bei verschiedenen Eingabewerten zu überprüfen und Randfälle abzudecken. Des Weiteren wurden zu den erstellen Use-Cases Testfälle erstellt um die wichtigsten Funktionen durch diese zu überprüfen. Die Testfälle beinhalten dabei ein Setup welches einen tatsächlichen Ablauf darstellt. Alle Testfälle wurden wie beschrieben durchgeführt.

5.1 Test zum Use Case Registrierung

Der erste Testfall dient zur Überprüfung des korrekten Verhaltens bei der Registrierung. Dabei ist anzumerken das die Bedingung dass es sich um eine @inf.uni-hannover.de handelt für diesen Testfall deaktiviert wurde. Der Testfall überprüft die korrekte Registrierung ohne Fehlermeldungen. Dieser wurde wie folgt durchgeführt:

Setup: Der Nutzer Finn Mikoschek öffnet den Webbrowser Chrome und öffnet <http://localhost:3000/register>. Der Nutzer möchte sich mit der Email Adresse `finnmikoschek@web.de` registrieren und dem Passwort 1.

Eingabe	Soll-Ausgabe
Nutzer gibt finnмикосчек@web.de ins Email Eingabe Feld ein und drückt den Button "Bestätigen"	Es erscheint der Text: Ihnen wurde ein Code an die angegebene Email geschickt, geben sie diesen bitte ein! und das Eingabe Feld ändert sich zum Code Eingabefeld. An die Email finnмикосчек@web.de wurde eine Email mit einem Zufälligen Code gesendet.
Nutzer gibt den Code aus der von seumfragetool@gmail.com gesendet Email ein und drückt den Button "Bestätigen"	Es erscheint der Info Text: Um die Registrierung abzuschließen geben sie bitte noch ein Passwort ein!, das Eingabe Feld ändert sich zu einem Passwort Eingabefeld, wodurch die Eingabe verpixelt wird.
Der Nutzer gibt das Passwort 1 ein und drückt den Button "Bestätigen"	Der Nutzer wird zur Startseite weitergeleitet. In welchem oben links Mikoschek angezeigt wird.

5.2 Test zum Use Case Frage erstellen

Der Testfall deckt den Use-Case zum Fragerstellen ab. Dabei werden die Latex-Funktion sowie das Löschen von bestimmten Antworten zusätzlich zu der Grundfunktion des Fragerstellens getestet. Dieser wurde wie folgt durchgeführt:

Setup: Der Nutzer Finn Mikoschek ist eingeloggt und möchte eine Frage in Latex darstellen. Welche vom Typen Single Choice ist. Die Frage soll 4 Antwort Möglichkeiten haben. Der Fragen Text ist Lösen sie folgende Formel $x = \sqrt{2^4 + 2^4 + 2^5}$. Die Antwort Optionen sind dabei 8, 16,4 und $\sqrt{8}$. Die erste Antwort ist dabei die Korrekte und die letzte Antwort soll ebenfalls in Latex dargestellt werden. Zudem wird aus Versehen eine fünfte Antwort erstellt, welche aber wieder gelöscht werden soll.

Eingabe	Soll-Ausgabe
Nutzer gibt Fragen Text: Lösen sie folgende Formel $x = \sqrt{2^4 + 2^4 + 2^5}$. ein und klickt die Checkbox Frage in Latex darstellen an und wählt als Fragen Typ Single Choice	Das Antwort Feld wird ausgeblendet und dafür wird ein Antworten Feld eingeblendet, welches ein Antwort Objekt beinhaltet. Des weiteren wird der Antwort hinzufügen Button sichtbar.
Nutzer drückt vier mal den Antwort hinzufügen Button.	Es werden 4 Antwort Objekte erstellt.
Nutzer füllt die Antwort Objekte wie folgt aus. Antwort 1: "8" Korrekt, Antwort 2: "16", Antwort 3: "4", Antwort 4 $\sqrt{8}$ Latex und löscht die 5 Antwort	Das Fünfte Antworten Objekt wird entfernt
Nutzer drückt Frage erstellen	Frage wird gespeichert und Es wird ein Alert Ihre Frage wurde erfolgreich erstellt! ausgegeben, des weiteren werden alle Felder auf den Standard wert zurück gesetzt.

5.3 Test zum Use Case Umfrage erstellen

Der Dritte Testfall testet ob das Umfrage erstellen funktioniert und überprüft des Weiteren ob bei nicht Eingabe notwendiger Parameter eine Fehlermeldung ausgegeben wird. Der Testfall wurde wie folgt durchgeführt:

Setup: Der Nutzer Finn Mikoschek möchte eine Umfrage erstellen mit dem Umfragen Titel Umfrage zur Vorlesung 1. Zu welcher die Frage Lösen sie folgende Formel $x = \sqrt{2^4 + 2^4 + 2^5}$. hinzugefügt werden soll. Die Umfrage soll eine Dauer von 100 Sekunden haben. Dabei vergisst der Nutzer den Umfrage Titel einzugeben und versucht die Umfrage zuerst ohne Titel fertigzustellen.

Eingabe	Soll-Ausgabe
Nutzer klickt die checkbox neben der Frage $x = \sqrt{2^4 + 2^4 + 2^5}$.	Die Fragen Anzahl Anzeige würd auf 1 gesetzt.
Nutzer gibt die Dauer von 100 Sekunden an und drückt auf Umfrage fertigstellen	Es wird ein alert Bitte geben sie einen Umfragen Titel ein und eine Umfragen Dauer an! dargestellt.
Der Nutzer gibt den Umfragen Titel Umfrage zur Vorlesung 1 ein und klickt auf Umfrage fertigstellen	Die Umfrage wird gespeichert und es wird der alert Ihre Umfrage wurde erfolgreich erstellt! ausgegeben

5.4 Test zum Use-Case Umfrage veröffentlichen

Der vierte Testfall überprüft ob die Veröffentlichung von vorab erstellten Umfragen funktioniert. Des Weiteren wird getestet ob die Umfrage in der Umfragenvorschau korrekt angezeigt wird. Der genannte Testfall wurde wie folgt ausgeführt:

Setup: Der Nutzer Finn Mikoschek möchte eine Umfrage mit dem Umfragen Titel Umfrage zur Vorlesung 1 veröffentlichen. Der Nutzer besitzt noch eine weitere Umfrage mit dem Titel Umfrage zur Vorlesung 2. Er möchte sich diese aber zuerst einmal in der Umfragen Vorschau anschauen und diese danach starten. Zudem möchte ein weitere Person an der Umfrage teilnehmen und tritt über den angezeigten Link bei.

Eingabe	Soll-Ausgabe
Nutzer klickt auf Umfragen - veröffentlichen	Die Umfragen "Umfrage zur Vorlesung 2" und "Umfrage zur Vorlesung 1" werden angezeigt.
Nutzer klickt auf "Umfrage zur Vorlesung 1"	Titel der Umfrage Fragen Anzahl Bearbeitungszeit sowie die Fragen von der geklickten Umfrage werden angezeigt.
Der Nutzer klickt auf Umfrage Vorschau	Ein neuer Tap mit der Umfragen Vorschau wird angezeigt
Der Nutzer schließt die Umfragen Vorschau und klickt auf "Umfrage vorbereiten"	Die Ansicht wird geändert. In der neuen Ansicht sind der link: http://localhost:3000/poll?pollid=61 Teilnahme Code 61 und ein QR-Code zusehen.
Der Umfragen Teilnehmer gibt den angezeigten Link " http://localhost:3000/poll?pollid=61 "	Es wird angezeigt das die Umfrage noch nicht gestartet wurde.
Der Umfragen ersteller klickt auf Umfrage starten	Die Umfrage wird gestartet und der Timer beginnt von 100 runterzuzählen. Zudem würd dem Teilnehmer nun die Umfrage angezeigt.

5.5 Test zum Use-Case Umfrageergebnisse Exportieren

Der letzte Testfall testet das Verhalten wenn eine Umfrage beendet wird und Umfrageergebnisse als PDF exportiert werden. Außerdem wird überprüft ob die Antworten korrekt dargestellt werden. Der genannte Testfall wurde wie folgt ausgeführt:

Setup: Der Nutzer Finn Mikoschek möchte die laufende Umfrage "Umfrage zu Vorlesung 1" beenden und die Umfrageergebnisse exportieren. An der Umfrage haben 2 Personen teilgenommen. Die Umfrage beinhaltet eine Frage mit 4 Antwort Optionen dabei ist die erste Antwort Option die richtige. Die erste Person hat die 2. Antwort Option als richtig gewählt und die zweite Person die erste Antwort Option.

Eingabe	Soll-Ausgabe
Nutzer klickt auf Umfrage beenden	Die Verbleibende Zeit wird auf 0 gesetzt und der "Umfrage Beenden" Button wird durch den "Umfragen Auswerten" Button ersetzt.
Nutzer klickt auf Umfrage Auswerten	Die in der Umfrage beinhaltete Frage wird als Balken Diagramm angezeigt welches auf der x Achse die Antwort Möglichkeiten zeigt und auf der y Achse die Teilnehmer Anzahl der Antworten
Der Nutzer klickt auf exportieren	Es wird eine Pdf zu der Umfrage erstellt welche ein Tabelle zu der Frage enthält. Es erscheint ein Fenster zum speichern der PDF
Der Nutzer speichert die PDF unter den Namen Umfrage zur Vorlesung 1 im Ordner downloads	Die Pdf wird im Ordner downloads gespeichert.

Kapitel 6

Verwandte Arbeiten

In diesem Kapitel wird die von mir entwickelte Plattform von anderen Umfrage Plattformen abgegrenzt. Eine zu betrachten Echtzeit-Umfrageplattform ist Pingo ([6]) welche ebenfalls während der Vorlesungen zum Einsatz kommt. Pingo ist allerdings darauf spezialisiert keine Umfragen, sondern einzelne Fragen während der Vorlesung zu stellen, weshalb diese Plattform keine Funktion bietet mehrere Fragen zeitgleich zu stellen und diese auszuwerten. Dabei liegt der Unterschied der Tools darin, das sich in dieser Arbeit beschriebene Tool mit Umfragen auseinandersetzt, welche aus mehreren Fragen besteht. [5].

Kapitel 7

Limitierung der Arbeit

Es wurde eine Echtzeit Umfrage Plattform entwickelt, welche aufgrund von zeitlicher Einschränkung nur Lokal getestet wurde. Dies könnte bei einer Testung über einen öffentlichen Server zu möglichen Problemen führen, da dieses Verhalten nicht getestet wurde. Des Weiteren ist zu betrachten, dass die Programmierung des Backends sowie des Frontends auf lediglich zwei Endgeräten ausgeübt worden ist und aufgrund der Verwendung eines Lokalen Servers nicht über ein Handy getestet werden konnte. Das könnte ebenfalls dazu führen, dass auf anderen Endgeräten Probleme auftreten könnten, da eine Testung über zwei Geräte bei einem solchen Tool nicht ausreicht. Zudem reichen die durchgeführten Tests dafür nicht aus das Umfrage Tool während einer Vorlesung durchzuführen. Dafür müsste das Tool erst einmal über verschiedene Geräte in Echtzeit getestet werden, da diese Testung des Echtzeit-Aspektes über einen Lokalen Server nicht vollständig möglich wäre.

Kapitel 8

Zusammenfassung und Ausblick

8.1 Zusammenfassung

In dieser Arbeit wurde eine interaktive Echtzeit-Umfrageplattform für Dozierende entwickelt und dokumentiert. Diese wurde durch eine Anforderungsanalyse und der verbundenen Umfrage geformt. Das Endprodukt bietet dabei die Grundlegenden sowie teilweise auch erweiterten Features an, die für eine Erstellung einer interaktiven Umfrage notwendig sind. Erweiterte Features sind dabei der Export der Umfrageergebnisse sowie dessen grafische Darstellung und die Darstellung mathematischen Notationen. Diese Features sind vor allem bei der Auswertung der Umfrage sehr nützlich. Allerdings gibt es noch mögliche Stellen zu verbessern, welche im folgenden Abschnitt erläutert werden.

8.2 Ausblick

Falls es zu einer Weiterentwicklung des Umfragetools kommen sollte, müsste man sich folgenden Punkten widmen: Die grafische Benutzeroberfläche müsste an verschiedene Bildschirmgrößen besser angepasst werden, um zu gewährleisten, dass alle Elemente gut sichtbar und gut bedienbar sind. In diesem Fall sollte es dann nämlich nicht dazu kommen können, dass der Text aufgrund von fehlendem Platz nicht vollständig angezeigt wird, ohne das Bedienelemente, wie Knöpfe, eine zu geringe Größe haben und somit schwer erkennbar sind. Des Weiteren müsste man sich der Implementierung auf einem Web Server und gegebenenfalls den damit verbundenen Problemen widmen.

Das Umfrage-Tool kann außerdem durch das Hinzufügen von weiteren Features erweitert werden. Diese könnten z.B. eine zeitlich gesteuerte Veröffentlichung einer Umfrage oder eine Umfrage ohne Zeitlimit sein. Das

Umfrage-Tool bietet zwar ein breites Repertoire an Funktionen, ist aber zum Einsatz während Vorlesung noch ungeeignet. Damit diesbezüglich mögliche Fehler vermieden werden könne, müsste es dafür zu einer größeren und gründlicheren Testung auf mehreren Geräten sowie verschiedenen Browsern kommen.

Literaturverzeichnis

- [1] A composable charting library built on React components. <https://github.com/recharts/recharts>. letzter Zugriff: September 2022.
- [2] A library to generate PDFs in JavaScript. <https://github.com/parallax/jsPDF>. letzter Zugriff: September 2022.
- [3] Django. <https://docs.djangoproject.com/en/4.1/>. letzter Zugriff: September 2022.
- [4] Javascript Bibliothek zur Erstellung von Interfaces . <https://reactjs.org/>. letzter Zugriff: September 2022.
- [5] Konzept von Pingo. <https://trypingo.com/de/faq/>. letzter Zugriff: September 2022.
- [6] Pingo Live-Feedback. <https://trypingo.com/de//>. letzter Zugriff: September 2022.
- [7] T. de Oliveira Cassiano. Display math in TeX with KaTeX and ReactJS. <https://github.com/talyssonoc/react-katex>. letzter Zugriff: September 2022.
- [8] J. McMahon. Authentication Module for Django rest auth. <https://github.com/James1345/django-rest-knox/>. letzter Zugriff: September 2022.
- [9] P. O'Shannessy. A React component to generate QR codes for rendering to the DOM. <https://github.com/zpao/qrcode.react>. letzter Zugriff: September 2022.
- [10] V. Pawlik. Umfrage in Deutschland zum Besitz von PCs und Computern nach Typ 2021. <https://de.statista.com/statistik/daten/studie/160925/umfrage/ausstattungsgrad-mit-personal-computer-in-deutschen-haushalten/>. letzter Zugriff: September 2022.

- [11] S.Bengtsson. Generate PDF tables with Javascript. <https://github.com/simonbengtsson/jsPDF-AutoTable>. letzter Zugriff: September 2022.
- [12] F. Tenzer. Anteil der Smartphone-Nutzer in Deutschland nach Altersgruppe im Jahr 2021. <https://de.statista.com/statistik/daten/studie/459963/umfrage/anteil-der-smartphone-nutzer-in-deutschland-nach-altersgruppe/>. letzter Zugriff: September 2022.