Informatikprojekt Kombinatorik-Trainer

Verbesserung und Erweiterung des Kombinatorik-Trainers

Autoren: Herr Tillmann Junk, Herr Fabian Kindgen

Im Auftrag von Prof. Dr. rer. nat. Wolfgang Konen

Inhaltsverzeichnis

[Inhaltsverzeichnis 1](#_Toc528578413)

[1. Projektplanung 2](#_Toc528578414)

[1.1. Altes System 2](#_Toc528578415)

[1.1.1 Parser 2](#_Toc528578416)

[1.1.2 (Aufgaben) 2](#_Toc528578417)

[1.2. Erweiterung Basiswissen 2](#_Toc528578418)

[1.2.1. Inklusion-Exklusion-Prinzip 2](#_Toc528578419)

[1.2.2. Binomialverteilung 3](#_Toc528578420)

[1.2.3. (Hypergeometrische Verteilung) 4](#_Toc528578421)

[1.2.4. Gemischte Aufgabentypen 5](#_Toc528578422)

[1.2.5. Dynamische Aufgabenerstellung 5](#_Toc528578423)

[1.3. Optionale Erweiterungen 8](#_Toc528578424)

[1.3.1. Feedback System (Würden wir als wichtig erachten) 8](#_Toc528578425)

[1.3.2. Studenten Interface für Ideen zu Aufgaben 8](#_Toc528578426)

[1.3.3. Einheitliches Interface zur Aufgabenerstellung 8](#_Toc528578427)

[2. Anhang 9](#_Toc528578428)

[2.1. Grobe Layouts 9](#_Toc528578429)

[2.1.1. Feedback abgeben 9](#_Toc528578430)

[2.1.2. Feedback Interface 9](#_Toc528578431)

[2.1.3. Studentenfragen Interface 10](#_Toc528578432)

[2.1.4. Aufgabenerstellung Interface 10](#_Toc528578433)

1. Projektplanung

**1.1. Altes System**

Im allgemeinem soll die Struktur der Website und der Funktionen beibehalten werden.

**1.1.1 Parser**

Der alte Parser soll soweit beibehalten werden, da ein Großteil der Fehler durch fehlerhafte Eingabeparameter der Parserfunktion entstehen. Die Parameter, um den Typ und die Kategorie einer zu spezifizieren, werden falsch übergeben. Dadurch geht die Parserfunktion in den falschen Zweig, welche dann ein Ergebnis in falscher Art und Weise berechnet.

Um weitere Fehler zu vermeiden bzw. besser analysieren zu können, würden wir ein System einbinden welches in Punkt „1.3.1. Feedback System“ beschrieben wird.

Der Parser für die Binomialverteilung soll eigenständig erstellt werden. Dieser wird modular aufgebaut sein, sodass dieser (oder Modulteile) für zukünftige Projekte weiter benutzt werden können. Dabei wäre es denkbar den alten Parser als Teilmodul einzubinden und damit weiter zu verwenden.

**1.1.2 (Aufgaben)**

Wir sind entschlossen die alten Aufgaben zu überabreiten und dynamisch generieren zu lassen, sofern dies im Zeitplan möglich ist und es als wichtig erachtet wird.

**1.2. Erweiterung Basiswissen**

Es wird auf die vorhandene Struktur der Website aufgebaut.

**1.2.1. Inklusion-Exklusion-Prinzip**

Verweis: Teschl Mathematik für Informatiker Band 1, Auflage 4, Kapitel 7, Seite 206/207

Vorstellung

Es soll unter dem Reiter Basiswissen auf der derzeitigen Website Informationen zum Inklusion-Exklusion-Prinzip angegeben werden. Dabei könnten Beispiele verwendet werden. Zudem könnte es einen eigenen Aufgabenbereich dazu geben.

Notizen

Für disjunkte Mengen gilt die Summenregel.

Für nicht disjunkte Mengen wird das Inklusions-Exklusions-Prinzip benutzt. Dabei wird die Anzahl der beiden Mengen summiert, um danach die doppelten/(N vorkommenden) Elemente einmal/(N -1 mal) zu entfernen.

|A u B| = |A| + |B| - |A n B| bei zwei Mengen

|A u B u C| = |A| + |B| + |C| - |A n B| - |A n C| - |B n C| + |A n B n C| bei drei Mengen

Benutzereingaben

Daraufhin könnten Benutzereingaben wie folgt eingegeben werden:

1. Das Ergebnis direkt als Zahl (z.B.: 1000)
2. oder als Formeln wie oben dargestellt (z.B.: |A| + |B| - |A n B|)
3. oder alles dazwischen. Sprich die Verwendung von Formeln und Zahlen.   
   (z.B.: 500 + 200 - |A n B|, (es gibt sicherlich ein besseres Beispiel))

**1.2.2. Binomialverteilung**

Verweis: Teschl Mathematik für Informatiker Band 2, Auflage 3, Kapitel 28.2, Seite 306-312

Vorstellung

Im Bereich Basiswissen wird um den Punkt Binomialverteilung erweitert. Dabei werden Informationen mit Beispielen angezeigt werden, um dem Benutzer das Thema näher zu bringen. Zudem wird es im Bereich Lernmodus einen eigenen Aufgabenbereich dazu geben.

Inhalt die unsere Meinung nach dargestellt werden sollten:

1. Bernoulli Experiment/Kette
2. Binomialverteilung
3. Eigenschaften der Binomialverteilung
   1. Erwartungswert und Varianz
   2. (Symmetrieeigenschaft)
   3. (Additionseigenschaft)
4. (Näherung der hypergeometrischen Verteilung)

Notizen

Eine Binomialverteilung entsteht, wenn wir uns für die Anzahl der Erfolge bei einer Bernoulli-Kette interessieren.

Gegeben ist eine Bernoulli-Kette der Länge n. Bei jeder der n Durchführungen kann ein bestimmtes Ereignis A mit der Wahrscheinlichkeit p eintreten (und das Gegenereignis -A mit der Wahrscheinlichkeit q = 1-p). Man interessiert sich für X = *Anzahl der Versuchsdurchführungen, bei denen A eintritt*. X kann die Werte x = 0,1,2,…,n annehmen. Die Wahrscheinlichkeit, dass A genau x-mal eintritt ist:

P(X = x) = (n über x) \* px \* qn-x

Erwartungswert E(X) = n\*p

Varianz Var(X) = n\*p\*q = n\*p\*(1-p)

Benutzereingaben

Daraufhin könnten Benutzereingaben wie folgt eingegeben werden:

1. Das Ergebnis direkt als Zahl (z.B.: 20% oder 0,20)
2. oder Zahlen vermischt mit Formeln. (z.B.: (7 über 3) \* (1/6)³ \* (5/6)5)

**1.2.3. (Hypergeometrische Verteilung)**

Verweis: Teschl Mathematik für Informatiker Band 2, Auflage 3, Kapitel 28.1, Seite 303-306

Vorstellung

Der Bereich Basiswissen könnte um den Punkt Binomialverteilung erweitert werden. Dabei werden Informationen mit Beispielen angezeigt werden, um dem Benutzer das Thema näher zu bringen. Zudem könnte es dann im Bereich Lernmodus einen eigenen Aufgabenbereich dazu geben.

Inhalt die unsere Meinung nach dargestellt werden sollten:

1. Hypergeometrische Verteilung
2. Erwartungswert und Varianz

Notizen

Gegeben ist eine Grundgesamtheit aus N Elementen, von denen M eine bestimmte Eigenschaft haben. Man entnimmt eine Stichprobe vom Umfang n (ohne Zurücklegen). Dann kann die Zufallsvariable X = Anzahl der Elemente in der Stichprobe mit der gewünschten Eigenschaft höchstens die Werte x = 0,1,2,…,n annehmen. Die Wahrscheinlichkeit, genau x Elemente mit der gewünschten Eigenschaft in der Stichprobe vorzufinden, ist

P(X = x) = (M über x) \* (N-M über n-x) / (N über n)

(für x > M bzw. n-x > N-M ist P(X = x) = 0 nach Definition des Binomialkoeffizienten – wir können nicht mehr blaue bzw. rote Kugeln ziehen als vorhanden sind)

Erwartungswert E(X) = n \* (M/N)

Varianz Var(X) = E(X) \* (1-E(X)) \* (N-n) / (N-1) = n \* (M/N) \* (1 - n \* (M/N)) \* (N-n) / (N-1)

Benutzereingaben

Daraufhin könnten Benutzereingaben wie folgt eingegeben werden:

1. Das Ergebnis direkt als Zahl (z.B.: 20% oder 0,20)
2. oder Zahlen vermischt mit Formeln. (z.B.: (4 über 2) \* (16 über 3) / (20 über 5))

**1.2.4. Gemischte Aufgabentypen**

Vorstellung

Der Prüfungsmodus könnte um die folgenden Punkte erweitert werden.

1. Den Prüfungsmodus erweitern, sodass auch Aufgaben der kommenden Erweiterung geprüft werden. Ziel wäre es dem Studierenden den Unterschied zwischen „Wann binomial, wann kombi „näher zu bringen.
2. (Prüfungsmodus mit einer Auswahl zu erweitern, um nur bestimmte Themen aktiv lernen zu können. Der Tester soll aus verschiedenen Kategorien und Bereichen Aufgaben wählen können.)

**1.2.5. Dynamische Aufgabenerstellung**

Hier wird im folgendem beschrieben wie die dynamischen Aufgaben in die Datenbank gespeichert werden müssen, um richtig vom PHP-Script interpretiert zu werden.

Mathematische Operationen:

|  |  |  |
| --- | --- | --- |
| Mathematische Operation | In der Datenbank | Im PHP Code |
| Addition | + | + |
| Subtraktion | - | - |
| Multiplikation | \* | \* |
| Division | / | / |
| Zuweisung | =…; | =…; |
| Klammersetzung | (…) | (…) |
| Modulo | % | % |
| Quadratwurzel | [sqrt]param01[\sqrt] | sqrt(param01) |
| N-te Wurzel als Potenz | [pow]base,expo[\pow] | pow(base,expo) |
| Potenz | Zahl^Exponent | \*\* |
| Zufall | [ran]min,max[ran] | rand(min,max) |
| Minimum | [min]param01,param02[\min] | min(p01,p02,…) |
| Maximum | [max]param01,param02[\max] | max(p01,p02,…) |
| Binomialkoeffizient | [binc]n,k[\binc] | binomial\_coeff(n, k) |
| Binomialverteilung | [bin]n,k,p[\bin] | binomialverteilung(n,k,p) |
| Summe Binomialverteilung | [sbin]n,k,p[\sbin] | summeBinomial(n,k,p) |
| Runden | [rnd]param01,Zahl[\rnd] | round(param01,Zahl) |
| Operation | [opr]…[\opr] | Nicht vorhanden |
| Parameter | paramXX | Nicht vorhanden |

Regeln der Mathematischen Operationen:

|  |  |
| --- | --- |
| Mathematische Operanten | Regel |
| Grundoperationen | |
| Addition | <ZAHL> v <FUNKTION> + <ZAHL> v <FUNKTION> |
| Subtraktion | <ZAHL> v <FUNKTION> - <ZAHL> v <FUNKTION> |
| Multiplikation | <ZAHL> v <FUNKTION> \* <ZAHL> v <FUNKTION> |
| Division | <ZAHL> v <FUNKTION> / <ZAHL> v <FUNKTION> |
| Zuweisung | <PARAMETER> = <ZAHL> v <FUNKTION>; |
| Klammersetzung | (<ANWEISUNG>) |
| Modulo | (<ZAHL> v <FUNKTION>) % (<ZAHL> v <FUNKTION>) |
| Funktionen | |
| Quadratwurzel | [sqrt] <ZAHL> v <FUNKTION> [\sqrt] |
| N-te Wurzel als Potenz | [pow] <BASIS>, <EXPONENT> [\pow] |
| Potenz | <BASIS>^<EXPONENT> |
| Zufall | [ran] <ZAHL> v <FUNKTION>,<ZAHL> v < FUNKTION > [\ran] |
| Minimum | [min] <ZAHL> v <FUNKTION>, <ZAHL> v <FUNKTION>, <ZAHL> v <FUNKTION>, …[\min] |
| Maximum | [max] <ZAHL> v <FUNKTION>, <ZAHL> v <FUNKTION>, <ZAHL> v <FUNKTION>, …[\max] |
| Binomialkoeffizient | [binc] <ZAHL> v <FUNKTION>, <ZAHL> v <FUNKTION> [\binc] |
| Binomialverteilung | [bin] <ZAHL> v <FUNKTION>, <ZAHL> v <FUNKTION>, <ZAHL> v <FUNKTION> [\bin] |
| Summe Binomialverteilung | [sbin] <ZAHL> v <FUNKTION>, <ZAHL> v <FUNKTION>, <ZAHL> v <FUNKTION> [\sbin] |
| Runden | [rnd]<ZAHL><RUNDUNGSSTELLE>[\rnd]  <RUNDUNGSSTELLE> ist definiert in:  …, -1 = Zehner, 0 = Einer, +1 = Zehntel, … |
| Operation | [opr]<ANWEISUNG>[\opr] |
| <BASIS> | <ZAHL> |
| <EXPONENT> | <ZAHL> |
| <FUNKTION> | Irgendeine Funktion die wiederum Funktionen aufrufen kann und einen Zahlenwert generiert. |
| <ANWEISUNG> | Zuweisung v <PARAMETER> |
| Parameter v <ZAHL> | Zahl mit Ziffern von 0-9 |

|  |
| --- |
| Beispiel |
| In der Datenbank |
| In einem Behälter befinden sich [opr]param03=param01+param02;[\opr] Kugeln, davon sind [opr]param01[\opr] blau und [opr]param02[\opr] rot. Aus dem Behälter werden nun ohne Zurücklegen [opr]param04=[ran][min]param01,param02[\min],[max]param01,param02[\max][\ran];[\opr] Kugeln zufällig entnommen. Wie groß ist die Wahrscheinlichkeit, in dieser Stichprobe genau [opr]param05=[ran]1,[min]param01,param04[\min][\ran];[\opr] blaue Kugeln vorzufinden? |
| Im PHP Code |
| In einem Behälter befinden sich param03=param01+param02; Kugeln, davon sind param01 blau und param02 rot. Aus dem Behälter werden nun ohne Zurücklegen param04=rand(min(param01,param02),max(param01,param02)); Kugeln zufällig entnommen. Wie groß ist die Wahrscheinlichkeit, in dieser Stichprobe genau param05=rand(1,min(param01,param04)); blaue Kugeln vorzufinden? |
| Ergebnis (Kann variiere, je nach Zufallszahl) |
| In einem Behälter befinden sich 11 Kugeln, davon sind 3 blau und 8 rot. Aus dem Behälter werden nun ohne Zurücklegen 4 Kugeln zufällig entnommen. Wie groß ist die Wahrscheinlichkeit, in dieser Stichprobe genau 2 blaue Kugeln vorzufinden? |

Parameterdefinition in der Datenbank:

Parameter (XX für Ziffern 0-9): paramXX

Zuweisungsparameter (XX für Ziffern 0-9): paramXX = …

Funktionsablauf:

1. Die Aufgaben werden aus der Datenbank geladen.
2. Das PHP-Script wandelt die Parameter und Funktionen in die für die Berechnung vorgesehene Operationen um und berechnet anschließend die Ergebnisse der Parameter und die Lösung der Aufgabe. Dabei geht es wie folgt vor:
   1. Als erstes werden alle Parameter, die keine Zuweisung sind, mit Zufallswerten initialisiert. Dabei kann ein Parameter verschiedenen Eigenschaften haben (z.B.: nur negativ, muss einen Wert zwischen X und y sein) die bei der Generierung der Zufallszahl eine Rolle spielen.
   2. Danach werden alle Zuweisungen initialisiert und berechnet. Dabei gelten die allgemeinen Rechen-regeln und -gesetzte wie oben definiert.
   3. Zu guter Letzt wird dann mithilfe des Lösungsschema die Lösung/(en) berechnet. (Wenn der Zeitplan es zulässt könnte man auch die Rechenschritte generieren und ausgeben lassen und/oder als PDF zum Download zur Verfügung stellen.)

Mehrere Fragestellungen und Lösungen

Mit den oben beschriebenen Regeln wäre eine Aufgabe mit mehreren Fragestellungen möglich. Man müsste nur das Layout der Website anpassen um mehrere Antworten zu ermöglichen oder man erstellt einen Parser welcher die Lösungen von einer Benutzereingabe sauber trennt und verarbeiten kann.

**1.3. Optionale Erweiterungen**

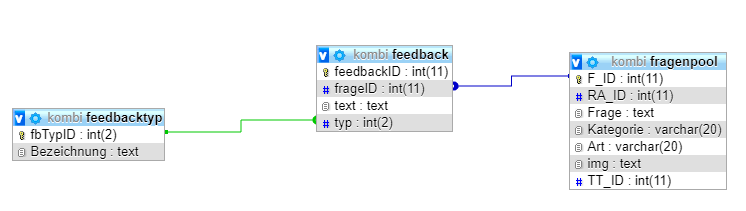
Im folgendem werden Ziele beschrieben die optional bzw. noch keine Einstufung haben.

**1.3.1. Feedback System (Würden wir als wichtig erachten)**

Ziel des Feedback-Systems wäre es, Fehler und Anmerkungen bzw. Verbesserungsvorschläge zu einer Aufgabe zu melden.

Dabei würde man eine DB-Tabelle erstellen, in der die Einträge gespeichert werden. Mithilfe eines Interfaces könnte man dann auf die Fehlermeldungen zugreifen.

ERD:



Ein grobes Layout sieht man im Anhang.

**1.3.2. Studenten Interface für Ideen zu Aufgaben**

Ziel währe es, dass Studenten eigene Ideen für Aufgaben oder komplette Aufgabenstellungen (inklusive Lösung) hinterlegen könnten, sodass man diese kontrollieren und ins System übernehmen kann. (Ggf. könnte man dafür Klausurpunkte für die Studenten vergeben).

Ein grobes Layout sieht man im Anhang.

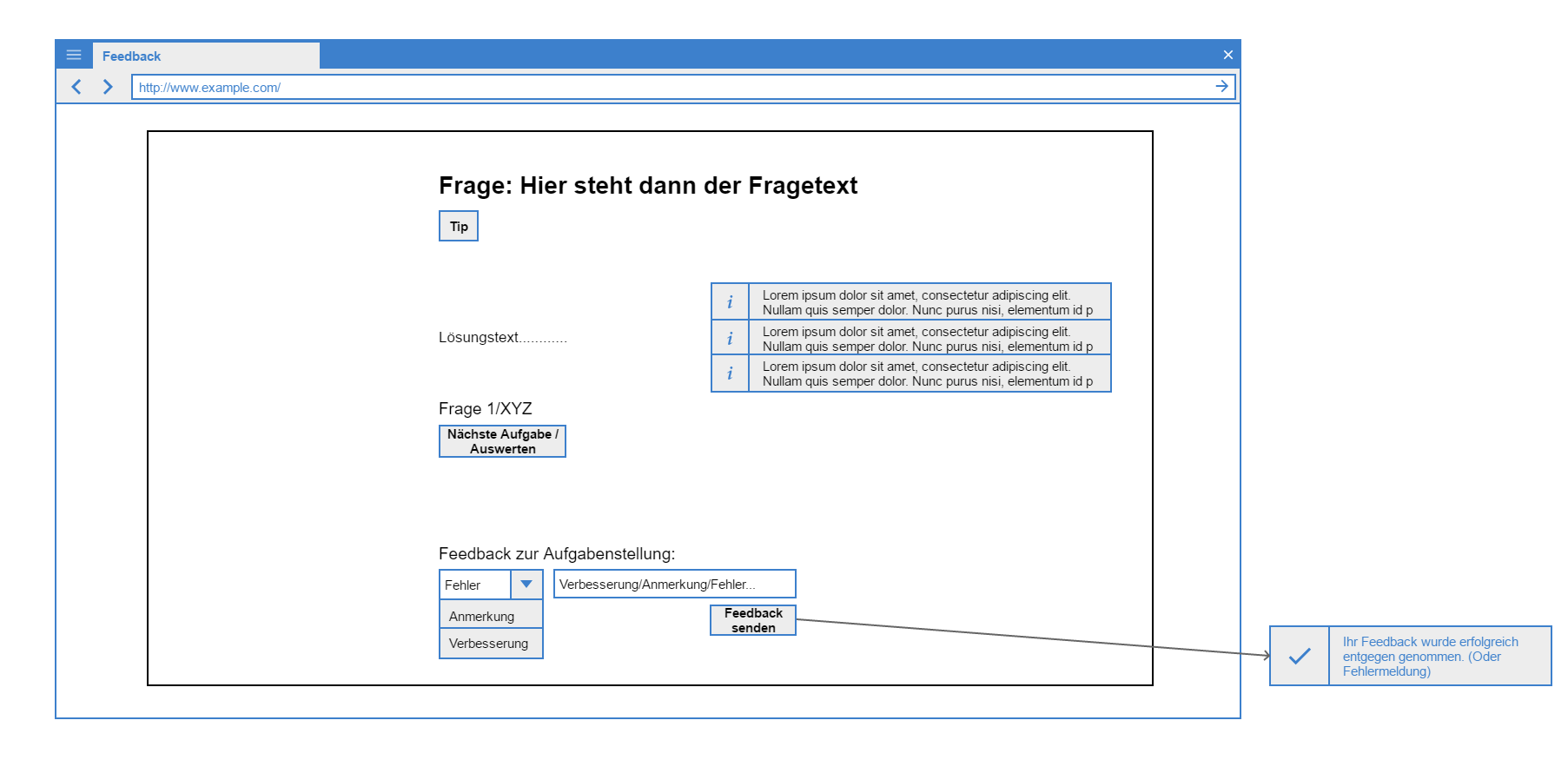
**1.3.3. Einheitliches Interface zur Aufgabenerstellung**

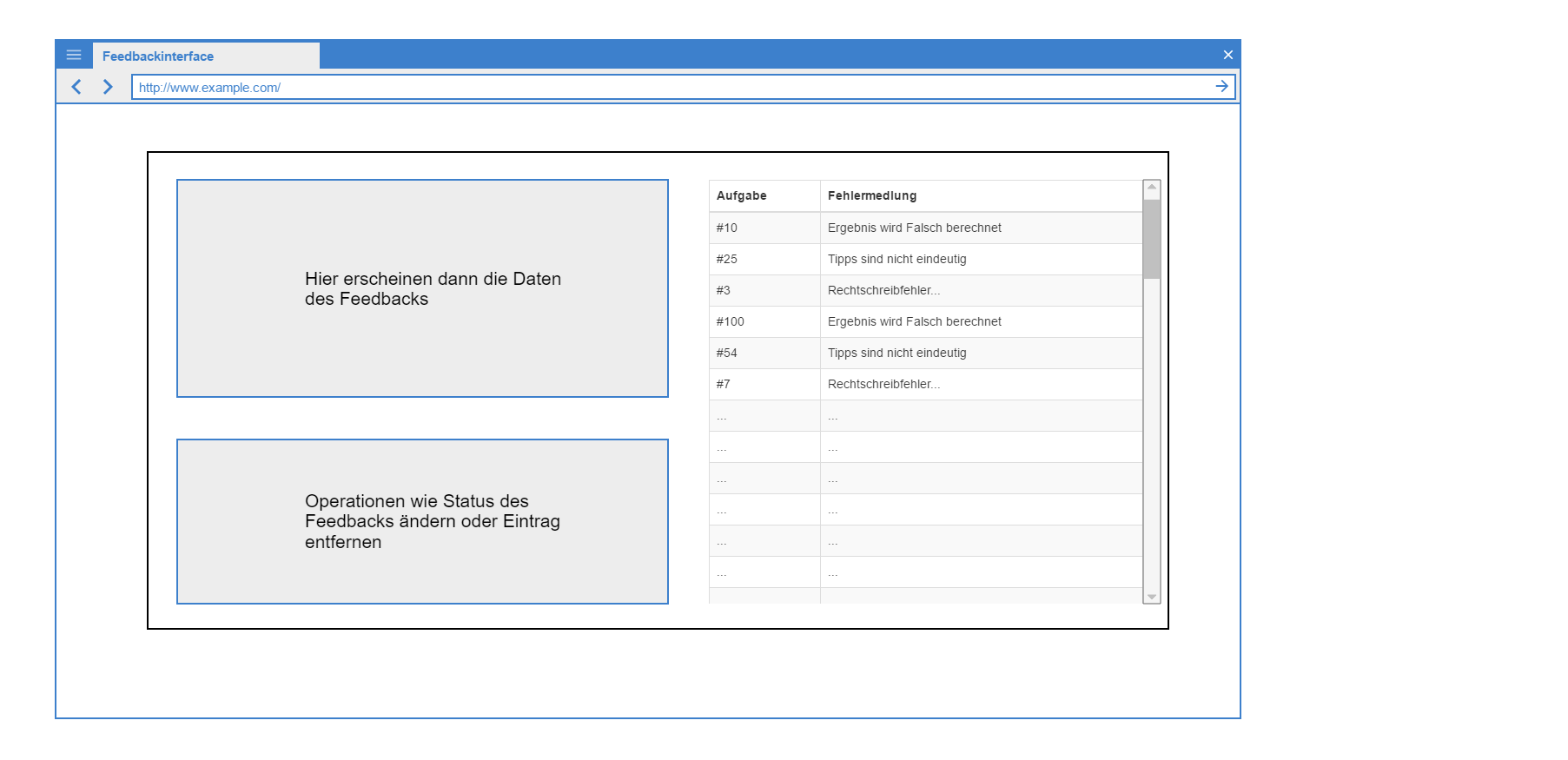
Damit man neue Aufgaben nicht umständlich über die DB eintragen muss, könnte man ein Interface entwickeln welche dann die Erstellung einer Aufgabe einfacher macht.

Ein grobes Layout sieht man im Anhang.

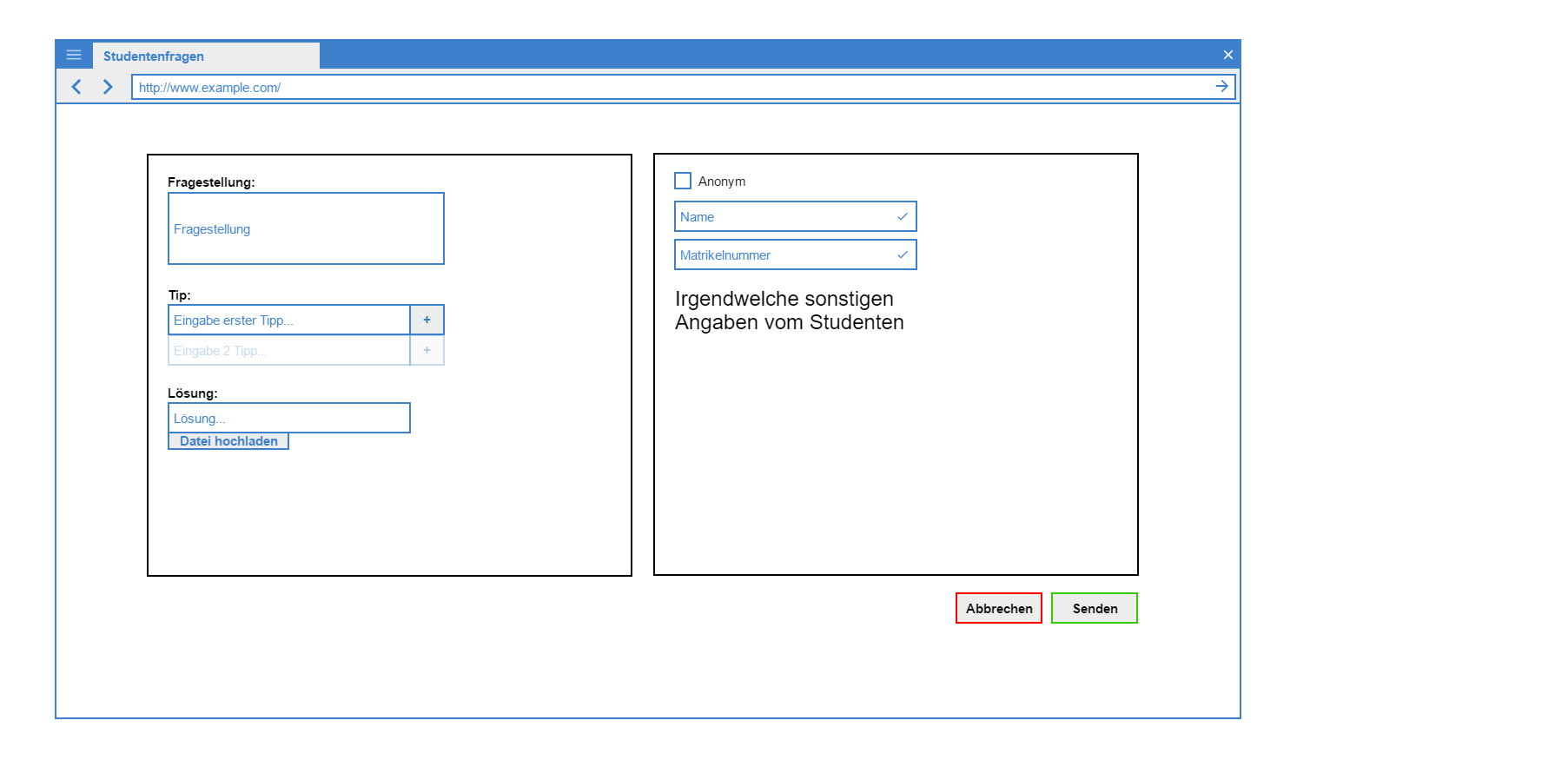
2. Anhang

**2.1. Grobe Layouts**

**2.1.1. Feedback abgeben**

**2.1.2. Feedback Interface**

**2.1.3. Studentenfragen Interface**



**2.1.4. Aufgabenerstellung Interface**