



**Universität
Zürich**^{UZH}

Bachelorarbeit am Institut für Informatik

28.05.2015

Webbasierte Re-Implementierung von LICARA nanoSCAN

Jennifer Studer

11-733-888

Visperterminen, Schweiz

Betreuender Professor: Prof. Dr. Lorenz Hilty

Betreuerin: Claudia Som, EMPA

Danksagung

Ich möchte allen danken, welche mich während der Erstellung dieser Arbeit unterstützt haben. Besonders danken möchte ich bei Claudia Som für die Betreuung dieser Arbeit und die zahlreichen freundschaftlichen Gespräche, die sie mit mir geführt hat. Grosser Dank gebührt auch Prof. Dr. Lorenz Hilty, welcher mir die Arbeit in seiner Forschungsgruppe ermöglicht hat. Die grosse Hilfe von Nicola Staub zu Beginn der Arbeit in die Einführung der Frameworks möchte ich hier nicht unerwähnt lassen. Zum Schluss geht noch ein herzlicher Dank an meinen Freund Raphael und an meine Familie, welche mich während dieser Zeit mit viel Energie unterstützt haben.

Abstract

Thanks to the expensive research the number of nanoparticles and nanomaterials and their variations have significantly increased during the last years. Even though great expectations are placed on nanoparticles there are still a lot of unanswered questions regarding the benefits and risks towards human and environment.

In September 2014, EMPA developed in cooperation with the companies TNO and NCB the LICARA guidelines to be used to conduct a Life Cycle, Risk and Benefit Assessment of nanoparticles. The LICARA nanoSCAN Excel version is connected to these guidelines and is acting as a semi-quantitative decision support tool while answering questions.

Within the scope of this bachelor thesis a web based version of the Excel tool has been developed to increase the usability of the software. The existing Excel tool has been analysed using the Design principles of Donald Norman. Based on these results the requirements were made. The developed tool has been tested by users and developers of the Excel tool and their feedback has been considered during the re-design.

Zusammenfassung

Die Anzahl der Nanopartikel und Nanomaterialien und deren Variationen ist in den letzten Jahren dank der Forschung stark gestiegen. Obwohl an diese Nanopartikel grosse Erwartungen gestellt werden, sind viele Fragen bezüglich den nanospezifischen Vorteilen und Risiken für Mensch und Umwelt noch unbeantwortet.

Zusammen mit den Unternehmen TNO und NCB hat die Empa im September 2014 die LICARA Richtlinien erarbeitet, welche bei der Einschätzung von Nutzen und Risiken solcher Nanopartikel für die Herstellung von innovativen Materialien, Produkten und Applikation helfen sollen. Dabei ist das LICARA nanoSCAN-Excel-Tool entstanden, welches durch Beantworten von Fragen entscheidungsunterstützend wirkt.

Im Rahmen dieser Bachelorarbeit soll die excelbasierte Version von LICARA nanoSCAN ins Web portiert werden, um die Benutzerfreundlichkeit der Software zu verbessern. Das bestehende Excel-Tool wurde anhand der Designprinzipien von Donald Norman analysiert. Aufgrund der Analyse wurden die Anforderungen an die Webapplikation erhoben und die dazugehörigen Designvorschläge kreiert. Nach der Implementierung wurde die Webapplikation mit Benutzern und Entwicklern der excelbasierten Version getestet. Die Erkenntnisse der Evaluation wurden bei einem Re-Design mitberücksichtigt.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Zielsetzung	1
1.2. Vorgehen	1
2. Hintergrund	3
2.1. Was ist LICARA nanoSCAN?	3
2.2. Verwandte Tools	4
3. Hauptteil	5
3.1. Analyse des Excel-Tools	5
3.1.1. Donald Normans Design-Prinzipien	5
3.1.2. Anwendung der Design-Prinzipien auf LICARA nanoSCAN	7
3.2. Definition der Anforderungen	10
3.3. Design	11
3.4. Implementierung	14
3.4.1. Technologie	14
3.4.2. Architektur	20
3.4.3. Datenmodell	23
3.4.4. Kommunikation	24
3.4.5. Benutzeroberfläche	25
3.5. Evaluierung	29
3.5.1. Feedback	29
3.5.2. Umsetzung	29
3.6. Schwierigkeiten	31
3.6.1. Cross-Browser-Kompatibilität	31
3.6.2. Analyse der Box-Kalkulationen	31
3.6.3. Balkendiagramme im PDF	31
4. Fazit und Ausblick	33
4.1. Fazit	33
4.2. Ausblick	33
Anhang A. Code für die Darstellung der Balkendiagramme bei unterschiedlichen Bildschirmgrößen	37
Anhang B. Code zum Download als Text-File	39

Tabellenverzeichnis

3.1. Analyse der HTML5 - Speichertypen	18
3.2. Übersicht der Rückmeldungen nach der Testphase	30

Abbildungsverzeichnis

2.1. Konzeptuelles Framework	3
3.1. Sichtbarkeit der Elemente im Google Kalender	6
3.2. Einkaufswagen auf amazon.com	7
3.3. Fehlermeldung bei falschem Eingabetyp	8
3.4. Überladung der Decision Support Box	9
3.5. Warnmeldung zur Irrelevanz der Box 6	9
3.6. Design der Haupt- und Boxennavigation	11
3.7. Design der Balkendiagramme	12
3.8. Design zur unterschiedlichen Darstellung von Input- und Drop- downfelder	12
3.9. Design zur übersichtlichen Darstellung der Decision Support Box	12
3.10. Design der Darstellung von gespeicherten SCANS	13
3.11. Anwendung der Technologien in der Architektur	14
3.12. Kollaboration von MVC Komponenten	14
3.13. Darstellung bei Bildschirmbreiten kleiner als 1200 Pixel	16
3.14. Darstellung bei Bildschirmbreiten grösser als 1200 Pixel	17
3.15. Support vom Download-Feld der einzelnen Browser	20
3.16. Klassendiagramm des JSON-Objektes boxes	24
3.17. Kommunikation beim Auswählen einer Antwort im Dropdown- Menu	25
3.18. Kommunikation beim Speichern eines Questionnaires	26
3.19. Hinweis auf fehlende Balkendiagramme	32

1. Einleitung

1.1. Zielsetzung

Ziel dieser Arbeit ist es, das auf Excel basierende Programm LICARA nanoSCAN ins Web zu portieren. Die Aufgaben zur Erreichung dieses Ziels sind die folgenden:

- Analyse der Schwachpunkte des bestehenden Excel-Tools, das durch Toon van Harmelen entwickelt wurde
- Ausarbeitung der Anforderungen
- Erstellung des Designkonzeptes
- Implementierung der Webapplikation
- Evaluierung und Redesign der webbasierten LICARA nanoSCAN Version

1.2. Vorgehen

In dieser Phase des Entwicklungsprozesses lag der Fokus auf das Einlesen in die Thematik, das konzeptionelle und inhaltliche Verstehen der LICARA Guidelines sowie das erste Kennenlernen der Excel-Version von LICARA nanoSCAN.

In der Analysephase wurde der Fokus vor allem auf die Schwächen hinsichtlich der Benutzerfreundlichkeit vom excelbasierten LICARA nanoSCAN gelegt. Hierfür wurde das Tool anhand den Design Prinzipien von Donald Norman analysiert.

Die designtechnischen Anforderungen wurden aufgrund der Ergebnisse der Analysephase festgelegt, die funktionalen und inhaltlichen Anforderungen hingegen in Zusammenarbeit mit Claudia Som.

In einem nächsten Schritt wurden die Designentwürfe für die jeweiligen Anforderungen konzipiert und ausgearbeitet.

In der Implementierungsphase wurde das Designkonzept umgesetzt.

In der letzten Phase des Entwicklungsprozesses wurde die webbasierte Version getestet und Feedback eingeholt. Die Rückmeldungen wurden in einem nächsten Schritt analysiert, woraufhin dementsprechend Designänderungen vorgenommen wurden.

2. Hintergrund

2.1. Was ist LICARA nanoSCAN?

LICARA ist die Abkürzung für 'Life Cycle Assessment and Risk Assessment of Nanoproducts', zu deutsch Lebenszyklus - und Risikobeurteilung von Nanoprodukten. LICARA nanoSCAN ist ein entscheidungsunterstützendes System für die Verwendung von Nanopartikeln und Nanomaterialien in der Herstellungsphase von Produkten, Applikationen und innovativen Materialien.

Obwohl an Nanopartikel und Nanomaterialien grosse Ansprüche gestellt werden, sind viele Fragen bezüglich den nanospezifischen Vorteilen und Risiken für Mensch und Umwelt noch unbeantwortet. Wie nachhaltig sind die Produkte? Wie sieht der rechtliche Sachverhalt aus? Welche Vorteile bringt eine Integration von Nanopartikeln in ein Produkt? Welche Risiken birgt der direkte Kontakt von Nanopartikeln mit Personen? Die Richtlinien im LICARA Konzept sollen kleinen und mittleren Unternehmen helfen diese Fragen zu beantworten. Sie basieren auf der wissenschaftlichen Arbeit der drei Forschungsinstituten TNO, Empa und RAS und den Erfahrungen von Unternehmen im privaten Sektor wie NCB, SNT, Freso, Nanothinx und AGPYME [Empa, 2014].

LICARA nanoSCAN ist modular aufgebaut und setzt sich aus insgesamt acht Boxen zusammen, welche in der Abbildung 2.1 dargestellt sind.

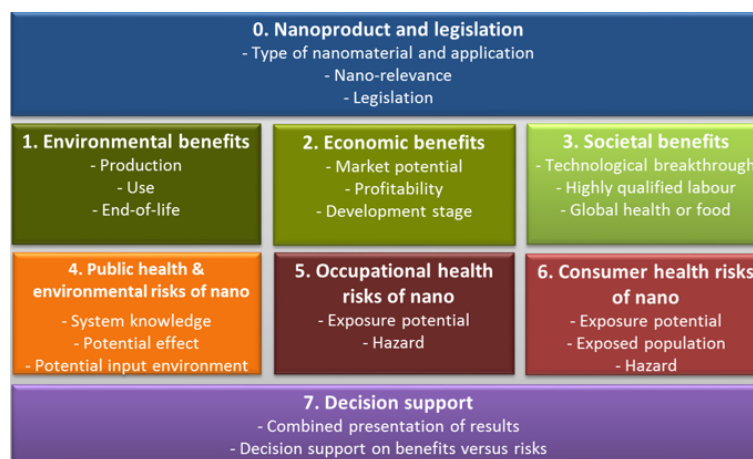


Abbildung 2.1.: Konzeptuelles Framework für LICARA nanoSCAN [Empa, 2014]

In der Box 0 'Nanoproduct and Legislation' wird anhand den Charakteristiken des Produktes berechnet, ob die Nutzung von LICARA nanoSCAN für

2. Hintergrund

das gewählte Nanoprodukt relevant und regelkonform ist. In den Boxen eins bis sechs werden die Vorteile und Risiken des Produktes im Vergleich zu einem konventionellen Produkt mit ähnlicher Funktionalität berechnet. Alle Boxen können unabhängig von einander ausgefüllt werden. In der letzten Box 'Decision Support' wird aufgrund der in den ausgefüllten Boxen berechneten Daten grafisch aufgezeigt, wie das Entscheidungshilfetooll den Einsatz des gewünschten Nanopartikels beurteilt: Das Tool kann empfehlen, noch mehr Forschung zu betreiben, mit der Nutzung des Produktes weiterzufahren oder abzubrechen oder auch darauf hinweisen, dass mehr Vorteile benötigt werden. Zudem kann das Tool auch ein unentschiedenes Resultat zeigen, wenn sich Risiken und Vorteile gegenseitig aufheben.

Um das gesamte existierende Wissen und die Erfahrung hinsichtlich der Beurteilung von Nanopartikeln zu nutzen, baut LICARA nanoSCAN auf mehreren Tools auf. Einige Fragen in den Boxen 'Nanoproduct and legislation' und in 'Public health and environmental risk' basieren auf der Precautionary Matrix [BAG, 2013]. Die Resultate des Online-Tools Stoffenmanager Nano [van Kronenburg 14, 2015] werden in der Box 5 'Occupational Risks' übertragen um daraus weitere Werte zu berechnen. Die Box 6 'Consumer Health Risk' basiert auf dem Tool NanoRiskCat [Hansen et al., 2013].

2.2. Verwandte Tools

In den vergangenen Jahren wurden einige Tools entwickelt, welche versuchen, die Risiken und Vorteile von Nanoprodukten zu eruieren oder deren Lebenszyklus zu beurteilen. Daraus sind Tools entstanden, welche sich dem sogenannten Control Banding verschrieben haben. Dabei handelt es sich um eine Risk-Assessment-Analyse, welche im Kontext von Unsicherheit ein Risikomodell anwendet. In diesem Modell ist das Risiko eine Funktion von der Stärke der Auswirkung (hazard) und der voraussichtlichen Wahrscheinlichkeit dieser Auswirkung (exposure). Unter anderem dienen die Tools Stoffenmanager nano und NanoRiskCat dem Control Banding, welche beide im LICARA nanoSCAN integriert sind. Das ökologische Risk Assessment wird von Tools wie IOMC Toolbox, ECOSAR oder EUSES genauer betrachtet. Die Programme CONSEXPO, RAPEX und IUCLID befassen sich mit dem Risk Assessment der menschlichen Gesundheit. Zudem sind Tools wie LCA2GO, SEEBALANCE und AGBALANCE entstanden, um den Lebenszyklus von Nanoprodukten zu beurteilen [Malsch et al., 2015].

All diese Tools bieten entweder Control Banding, Risk oder Benefits oder Lifecycle Assessments an, keines verfügt jedoch über eine Kombination aller vier Komponenten. LICARA nanoSCAN kann durch die Integration einiger dieser Tools die komplette Palette an verfügbaren Assessments-Methoden anbieten.

3. Hauptteil

3.1. Analyse des Excel-Tools

Zu Beginn der Arbeit sind typische Durchläufe der excelbasierten Version von LICARA nanoSCAN durchgeführt worden. Da das System nicht aufgrund fehlender inhaltlicher Probleme reimplementiert werden sollte, lag der Fokus der Analysephase vor allem auf dem Design. Um das User Interface hinsichtlich der Benutzerfreundlichkeit analysieren und bewerten zu können, wurden die Design-Prinzipien von Donald Norman zu Hilfe genommen.

3.1.1. Donald Normans Design-Prinzipien

1988 hat Donald Norman ein Buch geschrieben, welches auf die wichtigen Design-Prinzipien von Alltagsgegenständen eingeht. Warum viele Menschen häufig die falsche Herdplatte anmachen oder den falschen Knopf im Lift drücken, um die ursprünglich geplante Etage zu gelangen, liegt nicht an der Unfähigkeit dieser Menschen, sondern meist am schlechten Design der Gegenstände. Um diese Fehler einfacher anzugehen, hat Norman sechs fundamentale Prinzipien erarbeitet, auf welche während der Designphase eines Produktes eingegangen werden sollte, um ein gutes, für den Kunden einfach zu bedienendes Produkt zu entwerfen [Norman, 1988]. Obwohl das Buch bereits vor 27 Jahren erschienen ist, können die darin beschriebenen Richtlinien heute nicht nur für das Design von Alltagsgegenständen angewendet werden, sondern haben auch einen signifikanten Einfluss auf die Gestaltung von modernen Webseiten und Webapplikationen [McNeil, 2014].

Einfachheit Das Prinzip der Einfachheit bezieht sich auf die Darstellung von ähnlichen Funktionen auf eine ähnliche Art und Weise. Das Design von Operationen soll über das ganze System konsistent bleiben, dabei kann zwischen internen und externen Einfachheit unterschieden werden. Während bei der internen Einfachheit dieselben Regeln innerhalb der Webseite angewendet werden müssen, z.B. Styling der Buttons oder konsistente Position von Elementen mit derselben Funktion, sind bei der externen Einfachheit diese Regeln in anderen Applikationen ebenfalls wiederzufinden [McNeil, 2014]. So hat sich mit der Zeit das Schema entwickelt, dass auf Webseiten, welche von links nach rechts gelesen werden, das Logo oben links zu platzieren ist [Nielsen, 2001].

Eindeutigkeit Die Attribute eines Elementes, z.B. Form und Farbe, sollen signalisieren, wie man das Objekt zu bedienen hat. So deutet ein unterstrichenes Wort auf einer Webseite beispielsweise auf einen Hyperlink hin. Wechselt

3. Hauptteil

sich der Mauszeiger beim Überfahren eines Elements zu einer kleinen Hand, ist dies ein Zeichen, dass das Element nach Drücken der Maustaste gezogen werden kann. Dies sind Angewohnheiten, die in den Köpfen der Benutzer tief verankert sind und deswegen auch angewendet werden sollen. Die Benutzer sind sich der Bedeutung dieser Zeichen bewusst [McNeil, 2014].

Sichtbarkeit Ziel der Sichtbarkeit ist es, relevante Teile sichtbar zu machen und weniger wichtigere Elemente unauffälliger zu gestalten. Funktionalitäten sollten aufgrund ihrer Wichtigkeit ins Auge fallen. Der Schlüssel zum Erfolg liegt hier in der Priorisierung von Inhalten und Funktionen. Wird jedoch der gesamte Inhalt einer Seite als wichtig eingestuft, wird das Gegenteil des gewünschten Effekts erzielt und keine Elemente werden hervorgehoben. Die Herausforderung liegt hier in der Priorisierung von Elementen. Eine hilfreiche Frage dazu ist: Wenn der Benutzer nur eine einzige Aktion auf deiner Webseite ausführen könnte, welche wird dies sein? Wenn dies zum sichtbarsten und prominentesten Element auf der Webseite gemacht wird, wird der Benutzer diese Aktivität auch ausführen. Ein gutes Beispiel dazu ist die Online Version des Google Kalenders. Wird diesem eine unscharfe Ebene darübergelegt, sind die wichtigsten Aktionen, welche der Benutzer ausführen sollte, klar hervorgehoben [McNeil, 2014].



Abbildung 3.1.: Sichtbarkeit der Elemente im Google Kalender

Feedback Feedback ist ein weit verbreitetes Konzept in der Kontrollwissenschaft und IT. Feedback sind Informationen, welche das System dem Benutzer zurückschickt, was gerade getätigt wird und was daraus resultiert. Diese Rückmeldungen müssen unverzüglich nach der Bedienung des Gegenstandes bzw. der Handlung erscheinen, um die korrekte Kausalität zu gewährleisten. Man stelle sich eine Kommunikation vor, in der man seine eigene Stimme nicht hört. Oder ein Stift, der keine Spur hinterlässt. Feedback ist ein Mittel um Frust und Verwirrung beim Benutzer zu verhindern. Bereits kleine Rückmeldungen wie eine Vibration, ein Ton oder ein Farbwechsel von einem Element

zeigen dem Benutzer, dass seine ausgeführte Aktion eine Folgeaktion ausgelöst hat [McNeil, 2014].

Mapping Mapping ist ein technischer Begriff, der die Beziehung zwischen zwei Objekten beschreibt. Der Fokus liegt hier auf den Erwartungen des Benutzers hinsichtlich den Bedienelementen. Kann anhand der Bedienungselemente das richtige Ergebnis auf eine Aktion erwartet werden? Man spricht von einem natürlichen Mapping, wenn die Bedienungsvorrichtung eine äussere Analogie des benutzten Objektes darstellt. Als gutes Beispiel dient hier der digitale Einkaufskorb auf Onlineshops. Während man im realen Leben im Supermarkt mit einem Einkaufswagen seine Einkäufe erledigt, geschieht dies im World Wide Web per Mausklick. Die ausgewählten Einkäufe werden jedoch wie in der realen Welt ebenfalls in einen Einkaufskorb gelegt, was den Benutzern die Navigation erleichtert. Um die Benutzerfreundlichkeit zu steigern, sollen wenn immer möglich solche Symbole und Metaphern aus dem Alltag wiederverwendet werden [McNeil, 2014].



Abbildung 3.2.: Einkaufswagen auf amazon.com

Einschränkungen Einschränkungen innerhalb eines Systems oder einer Webseite dienen in einem gewissen Mass der Beschränkung von Interaktionen des Benutzers. Durch das Einschränken vom Benutzerverhalten wird der Zustand des Systems dem Benutzer vermittelt. So zum Beispiel deutet ein ausgegrauter 'Senden'-Button am Ende eines Formulars darauf hin, dass noch nicht alle Felder ausgefüllt sind, die zum Abschicken jedoch benötigt werden [McNeil, 2014].

3.1.2. Anwendung der Design-Prinzipien auf LICARA nanoSCAN

In den nächsten Abschnitten wird die excelbasierte Version anhand der zuvor erläuterten sechs Design Prinzipien von Donald Norman kritisch betrachtet.

Einfachheit Das Design der Applikation ist sehr simpel und ohne Überladung von Funktionalitäten aufgebaut. Bei den Hyperlinks taucht jedoch eine Inkonsistenz auf: In jeder Box wird der Benutzer per Mausklick auf den 'Weiter'-Link zur nächsten Box weitergeleitet, einen 'Zurück'-Link sucht der Benutzer jedoch vergeblich.

Eindeutigkeit Das Prinzip der Eindeutigkeit wurde bereits sehr gut umgesetzt. Hyperlinks, welche zu anderen Boxen führen, sind alle in blau gehalten

3. Hauptteil

und unterstrichen. Zudem wechselt der Mauszeiger vom Standard-Pfeil zum Zeigefinger-Symbol. Kann ein Wert in einem Dropdown-Menü ausgewählt werden, taucht beim Klicken auf die Zelle rechts ein kleiner Pfeil auf, welches das Symbol eines Auswahl-Menüs darstellt. Auf den ersten Blick sind diese Pfeile jedoch nicht ersichtlich, was die Benutzer dazu führen kann, die Frage mit einer eigenen Eingabe, und nicht mit einer im Dropdown-Menü verfügbaren Lösung zu beantworten. Bei eigener Eingabe wird der Benutzer mit der Fehlermeldung in Abbildung 3.3 darauf aufmerksam gemacht. Die Rückmeldung lässt nicht

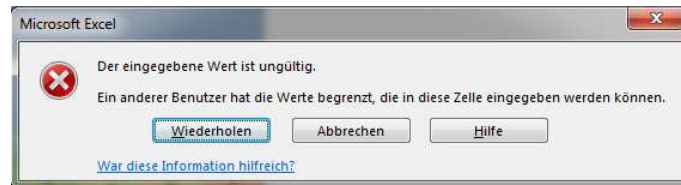


Abbildung 3.3.: Fehlermeldung bei falschem Eingabetyp

daraus schliessen, dass für das Beantworten der Frage das Dropdown-Menü benutzt werden soll. Diese Fehlinformation ist jedoch auf die limitierte Anzahl von Funktionen von Excel zurückzuführen.

Zudem kann ein kleines Pfeil-Symbol auf den 'Weiter'-Links unterstützend wirken.

Sichtbarkeit Die wohl wichtigste Box ist die letzte, in der die Entscheidungshilfe angeboten wird. Wie die Abbildung 3.4 zeigt, ist die Box jedoch mit viel Text überladen, was diese Seite für den Benutzer unattraktiv macht. Zu viel Text schreckt User vom Benutzen und Lesen ab, womit wichtige Informationen zur Interpretation der Grafiken verloren geht.

Feedback Wählt ein Benutzer eine Antwort in einem Dropdown-Menü aus, wird das normalisierte Resultat der ganzen Box neu berechnet. Dieser Wert wird zwar unterhalb der Box angezeigt, jedoch zieht ein einfacher Wertwechsel(z.B. die Zahl 0 wird durch 1 ersetzt) keine Aufmerksamkeit des Benutzers auf sich. Im Design des bestehenden Excel-Tools kann der Benutzer von dieser Funktionalität nur minimal profitieren.

Mapping Die Prinzipien des Mappings wurden sehr gut umgesetzt. Hier bedarf es keiner Verbesserung.

Einschränkungen Die Box 6 'Consumer Health Risk' ist von einer Frage der Box 0 'Nanoproduct' abhängig. Je nachdem, welche Antwort hier ausgewählt wird, ist die Box 6 relevant oder nicht und muss dementsprechend nicht ausgefüllt werden. Die Abbildung 3.5 zeigt, welcher Hinweis erscheint, falls diese Box irrelevant ist. Falls jemand jedoch trotz Warnmeldung, welche zwar dank der roten Farbe deutlich zu erkennen ist, die Box trotzdem ausfüllt, kann dies

3.1. Analyse des Excel-Tools

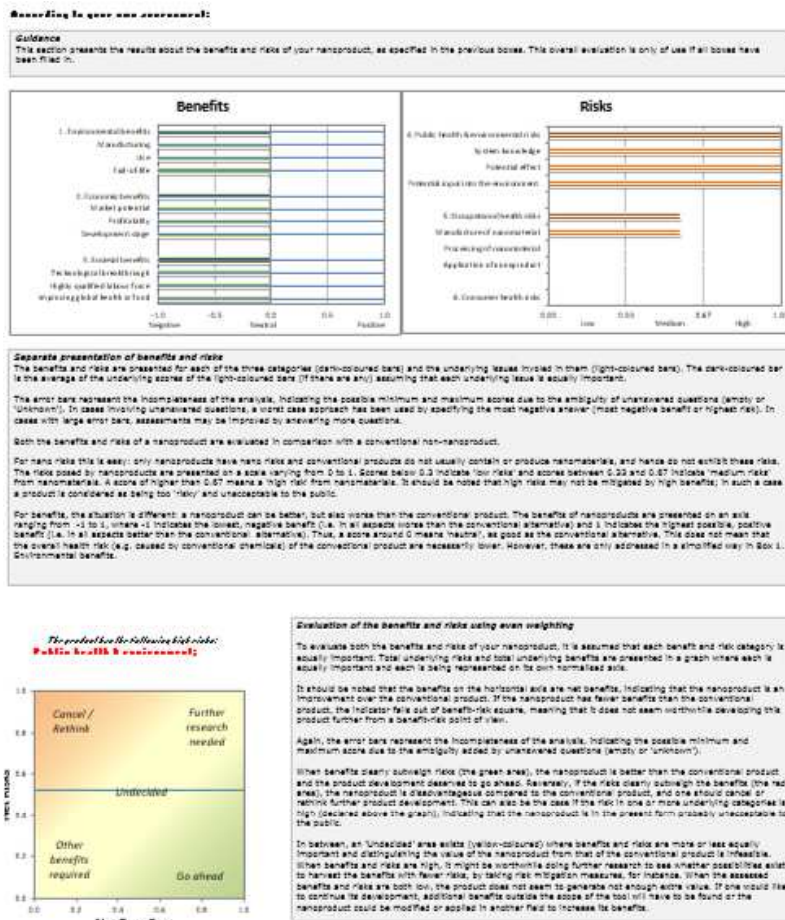


Abbildung 3.4.: Überladung der Decision Support Box

Box 6. Consumer Health Risks

Your product under consideration is (Q 0.4): An intermediate good
You can skip this sheet since consumers will not be involved with your product - please continue with 7. Decision making

Guidance
The questions in this box are only to be answered for consumer products or consumer/professional products. If your product does not involve consumers you can skip answering this box and proceed to Box 7 Decision support.

Abbildung 3.5.: Warnmeldung zur Irrelevanz der Box 6

in der letzten Box bei der Entscheidungshilfe zu Verwirrung führen. Die Box wird in diesem Falle nicht für die Berechnung der Entscheidung berücksichtigt - unabhängig davon, ob Daten eingegeben wurden oder nicht.

3. Hauptteil

3.2. Definition der Anforderungen

Im folgenden Abschnitt wird, basierend auf der Analyse der bestehenden Softwarelösung sowie der Wünsche von Claudia Som, auf die Anforderungen an die webbasierte Version eingegangen.

A1: 'Zurück'-Links hinzufügen Neben den 'Weiter'-Links sollen auch 'Zurück'-Links zur Verfügung stehen, um dem Benutzer die Möglichkeit zu geben, am Ende einer Box schnell und einfach auf die vorherige Box zu gelangen.

A2: Falsche Eingabe verunmöglichen Um dem Benutzer auf den ersten Blick zu zeigen, welche Felder ein Input-Feld und welche ein Dropdown-Menu sind, sollen beim Laden der Seite die Dropdowns durch den kleinen Pfeil direkt als solche ersichtlich sein.

A3: Icon zu 'Weiter'-Links und 'Zurück'-Links hinzufügen Die 'Weiter'-Links und 'Zurück'-Links sollen je mit einem passenden Icon ausgestattet werden.

A4: Übersichtlichkeit der Box 7 'Decision Support' gewährleisten Um die Übersichtlichkeit zu gewährleisten, soll der Text in eine Form gebracht werden, so dass sich User nicht mehr daran stört.

A5: Verbesserte grafische Darstellung des normalisierten Resultates Das normalisierte Resultat soll grafisch besser dargestellt werden, damit der User beim Beantworten einer Frage dessen Auswirkung auf das Resultat besser nachvollziehen kann.

A6: Ausfüllen der Box 6 'Consumer Health Risk' verunmöglichen bei Irrelevanz in Entscheidung Die Box 6 soll nicht ausfüllbar sein, wenn sie in der Entscheidung nicht mitberücksichtigt wird.

A7: Attraktives Layout Es soll ein attraktives Layout gewählt werden, beispielsweise wie beim Stoffenmanager nano oder bei der Precautionary Matrix, ohne die Einfachheit und Übersichtlichkeit zu verlieren

A8: Download als .txt oder .csv (Optional) Optional sollen die ausgefüllten nanoSCANs heruntergeladen werden, um sie zum Weiterschicken respektive zum Hochladen zur Verfügung zu stellen.

3.3. Design

In folgendem Abschnitt wird auf die Designvorschläge für die einzelnen Anforderungen eingegangen.

A7 spricht das attraktive Layout an, welches vom Aufbau her dem Stoffenmanager nano oder der Precautionary Matrix ähneln kann. Die Abbildung 3.6 zeigt, wie die Navigation aufgesplittet wurde. Links befindet sich die Hauptnavigation, bei welcher zwischen der Einführungsseite, den gespeicherten nanoSCANs sowie dem Start des nanoSCANs ausgewählt werden darf. Wählt der Benutzer den Menüpunkt 'Start nanoSCAN', gelangt er zu einer Ansicht mit einer zweiten Navigation. Diese dient zum Navigieren zwischen den einzelnen Boxen. Dieser Aufbau ist an denjenigen des Stoffenmanager nano Tools angelehnt.

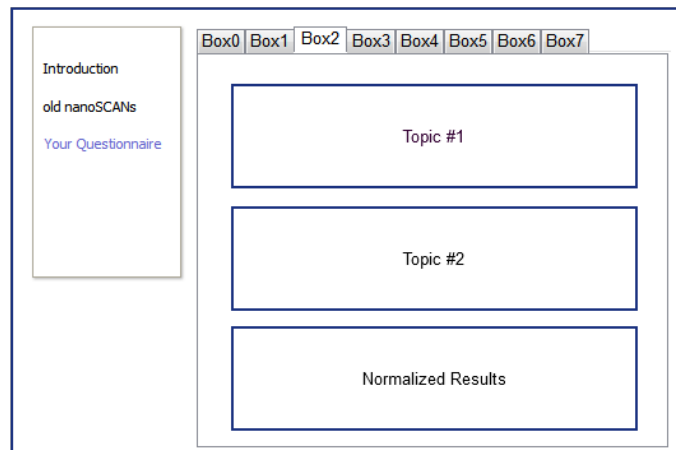


Abbildung 3.6.: Design der Haupt- und Boxennavigation

A1 und A3 sprechen die 'Weiter'- und 'Zurück'-Links in den Boxen an. Der 'Zurück'-Link sollte hinzugefügt werden und beide Buttons sollten mit einem Pfeil-Icon versehen werden, wie die Abbildung 3.7 zeigt.

Die Abbildung 3.7 zeigt zudem auch, wie A5 umgesetzt werden sollte. Die Werte pro Topic sollen in einem Balkendiagramm dargestellt werden. Durch die roten oder grünen Balkenfärbungen werden dem User eine zusätzliche Beurteilungshilfe angeboten. Die Balken sollen Farbe und Länge dynamisch anpassen: Beantwortet der User eine Frage, soll sich der Balken dementsprechend verschieben, vergrößern oder verkleinern und gegebenenfalls die Farbe ändern.

A2 spricht das Problem mit dem falschen Eingabetyp an. Beim Laden der Seite soll direkt ersichtlich sein, welche Frage mit einer eigenen Antwort und welche mithilfe eines Dropdown-Menüs beantwortet werden soll. Die Abbildung 3.8 zeigt, wie dies aussehen sollte.

A4 spricht die Unübersichtlichkeit der Box 7 'Decision Support' an. Hier sollte der Text in ein Akkordeon-ähnliches Element gepackt werden, um diesen falls gewünscht auf- oder zuzuklappen. Dadurch können die Grafiken mit oder

3. Hauptteil

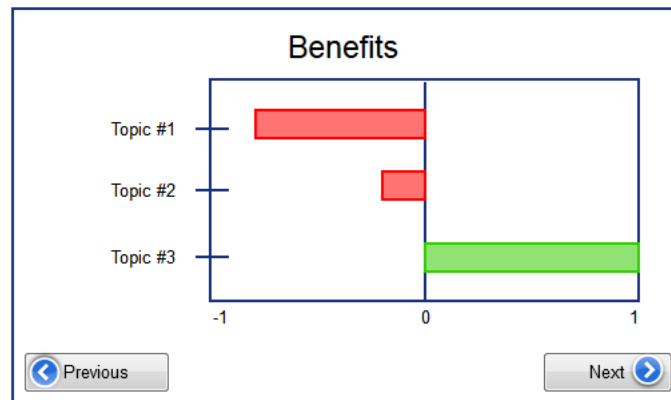


Abbildung 3.7.: Design der Balkendiagramme

Type of nano material and application

Q1.0 What is the main function that the nanomaterial provides in your application?

Q1.1 In which type of application is the nanomaterial be used?

Abbildung 3.8.: Design zur unterschiedlichen Darstellung von Input- und Dropdownfelder

ohne zusätzlichen Erklärungen betrachtet werden. Die Abbildung 3.9 zeigt den dazugehörigen Designvorschlag.

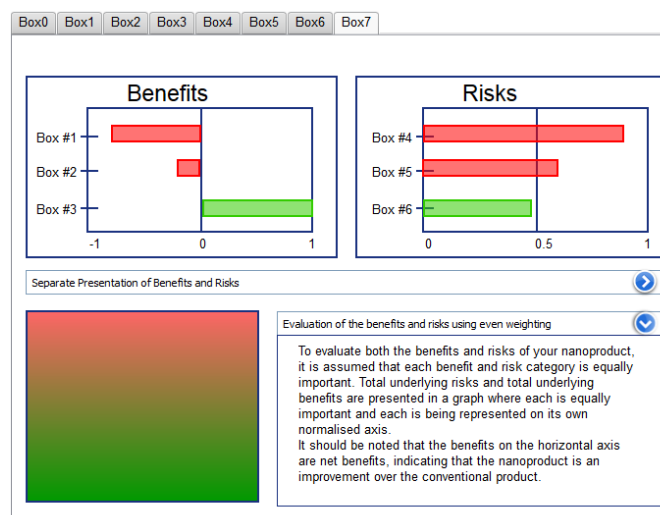


Abbildung 3.9.: Design zur übersichtlichen Darstellung der Decision Support Box

A6 spricht das Problem der Box 6 'Consumer Health Risk' an. Falls diese bei der Entscheidung irrelevant ist, sollen die Fragen dazu in dieser Box nicht angezeigt werden. Lediglich der Guidance-Text soll bestehen bleiben, welcher auf diese Thematik hinweist.

A8 spricht die optionale Download-Funktion eines nanoSCANs an. Die gespeicherten nanoSCANs sollten alle in einer eigenen Ansicht dargestellt werden. Übersichtshalber sollte auch hier, wie in der 'Decision Support' Box, auf die Akkordeon-ähnlichen Elemente zurückgegriffen werden. Klickt der Benutzer auf ein nanoSCAN, sollen seine Grafiken angezeigt werden. Wie die Abbildung 3.10 zeigt, sollte er hier zudem die Möglichkeit erhalten, den nanoSCAN herunterzuladen (als PDF und TXT), zu bearbeiten und zu löschen.

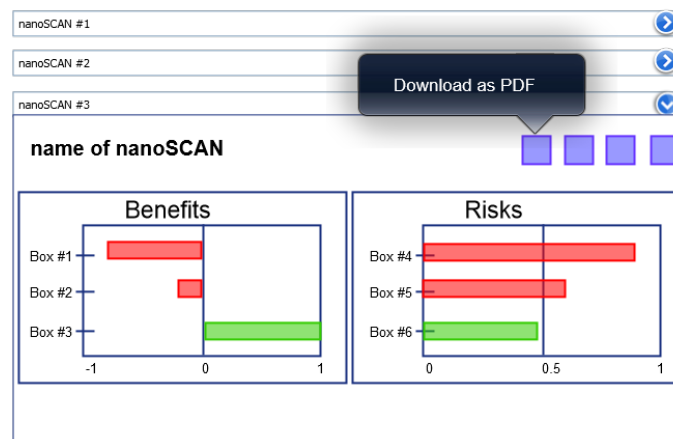


Abbildung 3.10.: Design der Darstellung von gespeicherten SCANs

3. Hauptteil

3.4. Implementierung

In diesem Kapitel wird in einem ersten Schritt auf die eingesetzten Technologien eingegangen. Zudem wird aufgezeigt, wie die Architektur und das Datenmodell aussehen und wie die einzelnen Einheiten der Webapplikation zusammenspielen. Die Benutzeroberfläche wird im letzten Schritt thematisiert.

3.4.1. Technologie

Im folgenden Abschnitt werden die wichtigsten Frameworks und ausgewählten Technologien beschrieben, welche bei der Entwicklung der Webapplikation eine wichtige Rolle gespielt haben. Die Abbildung 3.11 zeigt auf, wo die jeweiligen Technologien in der Architektur zu finden sind.

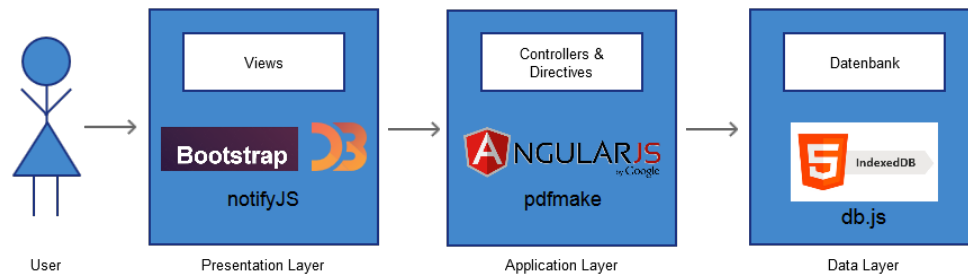


Abbildung 3.11.: Anwendung der Technologien in der Architektur

Web Framework

AngularJS AngularJS ist eine Open-Source-Software von Google und dient der Entwicklung von Webapplikationen [Google, 2015]. Es ist ein clientseitiges Framework, welches statische HTML Seiten mit einem klassischen Model-View-Controller (MVC) ausstattet und dadurch dynamisch macht. Im Architekturpattern Model-View-Controller wird die Applikation in die drei Einheiten Datenmodell (Model), Präsentation (View) und Steuerung (Controller) unterteilt.

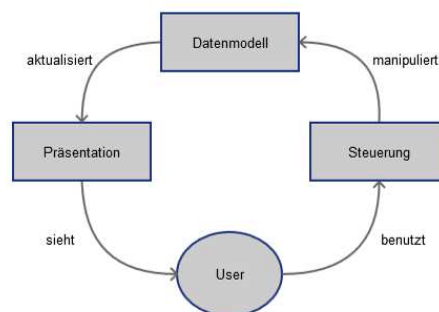


Abbildung 3.12.: Kollaboration von MVC Komponenten

3.4. Implementierung

Die Steuerung und die Präsentation implementieren das klassische Strategie-Pattern: Die Präsentationsschicht ist ein Objekt, welches mit einer Strategie konfiguriert ist. Diese Strategie wird in der Steuerung bestimmt. Die Präsentation fokussiert sich nur auf die visuellen Aspekte der Applikation. Zudem gibt sie der Steuerung die Aufgabe, Entscheidungen über das Interface-Verhalten zu treffen. Dieses Design Pattern ermöglicht es zudem auch, das Datenmodell von der Präsentation loszukoppeln, da es die Aufgabe der Steuerung ist, auf Userinteraktionen mit dem Modell zu interagieren. Die Präsentationsschicht hat keinen Einblick, wie die Daten, welche sie darstellt, aufbereitet wurden. [Freeman et al., 2004] AngularJS kommt mit vielen, gut zusammenspielenden Frameworks daher. Ein grosser Nachteil des Einsatzes ist jedoch, dass dabei die Unabhängigkeit verloren geht. Wird AngularJS in einer zukünftigen Version auf Funktionalitäten verzichten bzw. nicht mehr unterstützen, welche für LICARA nanoSCAN eingesetzt wurden, müsste die Applikation angepasst werden, um mit der neuen Version fehlerfrei zu laufen.

Darstellungs-Frameworks Im folgenden wird auf ausgewählte Frameworks eingegangen, welche bei der Präsentation von Daten in der Webapplikation eine wichtige Rolle spielen.

Bootstrap Bootstrap ist eine von Twitter entwickelte Open-Source-Software und ist das am meisten verarbeitete CSS-Framework, welches den Programmierer bei der Darstellung von Inhalten im Web unterstützt [Mark Otto, 2015]. Zudem kann die zusätzliche Javascript-Bibliothek heruntergeladen werden, um auf Interaktionen des Users zu reagieren, wie zum Beispiel für akkordeon-ähnliche Auflistung von Inhalten, welche für die Darstellung der alten nanoSCANs benutzt wurden. Bootstrap kommt mit einer umfangreichen Auswahl von CSS-Klassen daher, welche die optimale Darstellung von Inhalt auf Geräten mit unterschiedlich grossen Displays erheblich erleichtern - eine Herausforderung, welche mit den heute unterschiedlichsten Devicetypen und -grössen ohne Hilfsframeworks kaum zu bewältigen ist. Die Funktionalität, von welcher für diese Arbeit am meisten Gebrauch gemacht wurde, ist das 12-Gitter System. Die ganze Breite einer Webseite wird in zwölf Kolonnen unterteilt. Möchte man für einen Bereich, z.B. die Navigation nur einen Drittel der gesamten Breite verwenden, kann dies wie folgt angegeben werden:

```
<div class="row">
  <div class="col-xs-3">Platzhalter fuer die
    Navigation
  </div>
</div>
```

Die Klasse `col-xs-3` wird auf Geräten mit einem Bildschirm kleiner als 768px ein Drittel der verfügbaren Breite einnehmen.

3. Hauptteil

```
<div class="row">
  <div class="col-xs-3 col-lg-5">Platzhalter
    fuer die Navigation
  </div>
</div>
```

Die Navigation wird durch die Klasse `col-xs-3` auf Geräten mit einem Bildschirm kleiner als 1200 Pixel einen Drittel, ansonsten $5/12$ der verfügbaren Breite einnehmen. Diese Methode wird für die Balkendiagramme in den einzelnen Boxen angewendet, Anhang A zeigt den dazugehörigen Code. Bei Bildschirmen kleiner als 1200px werden die beiden Balkendiagramme für Vorteile und Risiken je die gesamten zwölf Kolonnen einnehmen. Bei Screens, welche

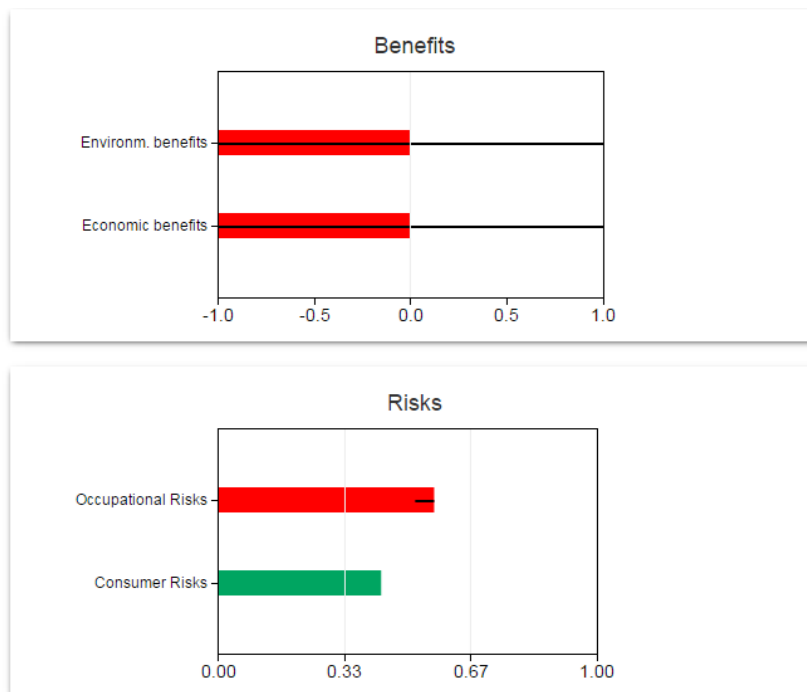


Abbildung 3.13.: Darstellung bei Bildschirmbreiten kleiner als 1200 Pixel

breiter als 1200px sind, werden diese nebeneinander dargestellt. Durch die Klasse `col-lg-offset-1` wird zwischen den zwei Diagrammen eine Kolonne ausgelassen, was in den Abbildungen 3.13 und 3.14 ersichtlich ist.

D3.js D3.js ist eine Javascript-Bibliothek, welche das grafische Visualisieren von Daten unterstützt [Bostock, 2013]. Tests haben gezeigt, dass bei der Anwendung von D3.js die Webseite dreimal schneller lädt als bei Verwendung von Flash [Bostock et al., 2011]. Aus diesem Grund wurde für die Generierung der Barcharts und des Koordinatensystems in der Decision Support View dieses Framework eingesetzt.



Abbildung 3.14.: Darstellung bei Bildschirmbreiten grösser als 1200 Pixel

notifyJS notifyJS ist ein jQuery Plugin, mit welchem sehr einfach Benachrichtigungen generiert werden [Pillora, 2015]. Diese Benachrichtigungen werden angezeigt, wenn der User aufgrund seiner Interaktion ein Feedback des Systems erwartet. So wird oben rechts in der Webapplikation eine Benachrichtigung angezeigt, wenn ein nanoSCAN beispielsweise erfolgreich oder erfolglos gespeichert oder gelöscht wurde.

Datenbank Wenn Webentwickler an Datenspeicherung denken, wird sofort an das Hochladen auf einen Server gedacht. Dank HTML5 gibt es jedoch zahlreiche neue Technologien, welche das Speichern von Daten auf dem Client ermöglichen. Diese Daten können, falls gewünscht, mit einem Server synchronisiert werden, können jedoch auch nur auf dem Client verfügbar bleiben.

Beim Projekt LICARA nanoSCAN ist es wichtig, dass keine Daten an einen Server vermittelt werden, damit nicht später gegen die Hersteller vorgegangen werden kann, wenn Nanopartikel trotz negativer Einschätzung durch LICARA nanoSCAN bei der Produktentwicklung eingesetzt wurden. Aus diesem Grund galt es, eine Speichermöglichkeit zu finden, damit nanoSCANs früherer Sessionen nach Aufrufen der Webseite bei Bedarf angezeigt, bearbeitet und heruntergeladen werden können. Eine Analyse der verfügbaren Speichertypen ergab die Übersicht in der Tabelle 3.1.

Aufgrund dieser Analyse wurde für LICARA nanoSCAN der Speichertyp IndexedDB gewählt. Zudem wird vom World Wide Web Consortium W3C die Nutzung von IndexedDB empfohlen, da dies der neue Standard der Datenspeicherung werden soll [Nikunj Mehta, 2015]. Eine indexierte Datenbank ist eine Kollektion von Objektspeichern, in welche ganz einfach Objekte gespeichert werden können. Die einzelnen Objektspeicher können mit SQL Tabellen verglichen werden, wobei hier keine Einschränkungen zur Objektstruktur definiert werden. Um jedoch nicht auf die unterste Ebene direkt zugreifen zu müssen, wurde das Framework DB.js ausgewählt. DB.js ist ein Wrapper für IndexedDB, der das Nutzen dieses Speichertyps stark vereinfacht [Powell, 2012].

Mehrsprachigkeit Eine Anforderung an die webbasierte LICARA nanoSCAN Version war, dass die Webseite in mehreren Sprachen verfügbar ist. Das angular-translate-Modul wurde eingesetzt, um die Mehrsprachigkeit zu ermöglichen. Die einzelnen Strings sind in verschiedenen Files im Verzeichnis /resources/

3. Hauptteil

Kriterium	LocalStorage	SessionStorage	IndexedDB	Web-SQL
Persistent	Ja	Nein	Ja	Ja
Asynchron	Nein	Nein	Ja	Ja
Limit	Ja. Zwischen 5MB - 25MB	Ja. Zwischen 5MB - 25MB	Ja. Browser- Abhängig. Browser fragt um Benutzer- erlaubnis, um mehr Daten zu speichern	Ja. Browser- Abhängig. Browser fragt um Benutzer- erlaubnis, um mehr Daten zu speichern
Performance	Schlecht, da nur 'key-value'- Typen gespeichert werden. Keine Inde- xes erlaubt.	Schlecht, da nur 'key- value'-Typen gespeichert werden. Keine Inde- xes erlaubt.	Sehr gut. Zwar ein- facher 'key-value'- Speicher, aber Inde- xes werden erlaubt. Gute Such- performance.	Sehr gut. Aufbau einer relationalen Datenbank. Gesucht kann via Search-Keys.
Browser- Support	Sehr guter Desktop - & Mobile- Support	Sehr guter Desktop - & Mobile- Support	Sehr gut in allen modernen Browsern. Für älte- re gibt es Polyfills	Noch sehr gut. Wird jedoch von neuen Brow- sern nicht mehr unter- stützt.
Bemerkungen		Daten wer- den beim Schliessen des Browser- Fensters gelöscht.	Der neue Standard.	Veraltet.

Tabelle 3.1.: Analyse der HTML5 - Speichertypen [Mahemoff, 2010]

unter `local-languagetag.json` gespeichert, wobei `languagetag` mit dem jeweiligen Sprachenschlüssel zu ersetzen ist. Fügt man einem HTML-Tag das Attribut `translate` hinzu, wird automatisch der String aus dem aktuell ausgewählten JSON geladen und in die View eingefügt. Klickt der User auf ein Sprachen-Icon im Header, wird die Funktion `changeLanguage` aufgerufen, welche per Translation-Service die Sprache geändert. Der Translation-Service informiert alle HTML-Tags, welche das Attribut `translate` enthalten, um aus dem neuen JSON die dazugehörigen Strings auszulesen und einzufügen.

```
$scope.changeLanguage = function (langKey) {  
    $translate.use(langKey);  
};
```

Entwicklungsumgebung LICARA nanoSCAN wurde in einer Windows 7 VM entwickelt, welche mit Java 1.8 ausgerüstet ist. Als IDE wurde Webstorm ausgewählt, da Webstorm sehr gute Javascript-Funktionalitäten zur Verfügung stellt, wie zum Beispiel Autovervollständigung, welche für Javascript von den meisten IDE's nicht ausreichend unterstützt wird. Um die Webapplikation lokal laufen zu lassen, wurde ein NodeJS-Server eingesetzt. NodeJS ermöglicht es sehr einfach schnell und skalierbare Webapplikationen zu bauen [Chaniotis et al., 2014]. Zudem wurde die Entwicklungsumgebung mit Grunt und Bower ausgerüstet. Diese beiden Bibliotheken haben zahlreiche Module, welche die Webentwicklung erleichtern, in dem sie automatisierte Aufgaben übernehmen. Zum Beispiel hat sich ein Modul um den automatisierten Pageload gekümmert, sobald ein JavaScript-Dokument oder ein HTML-Dokument aktualisiert wurde.

Cross-Browser-Kompatibilität Eine Webseite, welche für verschiedene Browser entwickelt wurde, trägt den Namen Cross-Browser Webseite. Der Job eines Webentwicklers wäre viel einfacher, wenn sich alle Browser gleich verhalten würden. Während die meisten im Design und den Funktionen ähnlich sind, gibt es trotzdem oftmals kleine Unterschiede in der Darstellung von Webseiten. Während die einen Browser die neuesten HTML-Tags unterstützen, werden diese von älteren Browsern noch nicht erkannt und demnach nicht korrekt interpretiert [Choudhary et al., 2010]. Dasselbe Problem taucht mit JavaScript auf: Ein Skript kann in einem Browser problemlos laufen, in einem anderen jedoch werden Fehler geworfen [Christensson, 2009]. Weiter kommt dazu, dass die Darstellung einer Webseite im Firefox auf einem Windows-Rechner nicht dieselbe sein muss wie auf einem Linux-Rechner.

Aus diesem Grund ist das Testing in den verschiedenen Browsern unabdingbar. Auch während dem Entwickeln von LICARA nanoSCAN tauchten Unterschiede in der Unterstützung der ausgewählten Technologien auf. Beispielsweise soll der User, als optionales Feature, einen ausgefüllten nanoSCAN als .txt-Dokument herunterladen können.

3. Hauptteil

Der Browser Chrome ist dafür bekannt, immer auf dem neusten Stand der Technologien zu sein und den Entwicklern dadurch den Umfang von Funktionalitäten von HTML5 und CSS3 in kürzester Zeit zur Verfügung zu stellen. So kann ein Dokument bei Klick auf einen Link direkt heruntergeladen werden, wenn das HTML-Tag das Feld `download` enthält. Von welchen Browsern dieses Feld interpretiert wird, zeigt die Webseite www.caniuse.com auf, welche den Support von HTML5-Feldern und JavaScript-Funktionalitäten der einzelnen Browsern darstellt. Die Abbildung 3.15 zeigt, dass das Download-Attribut

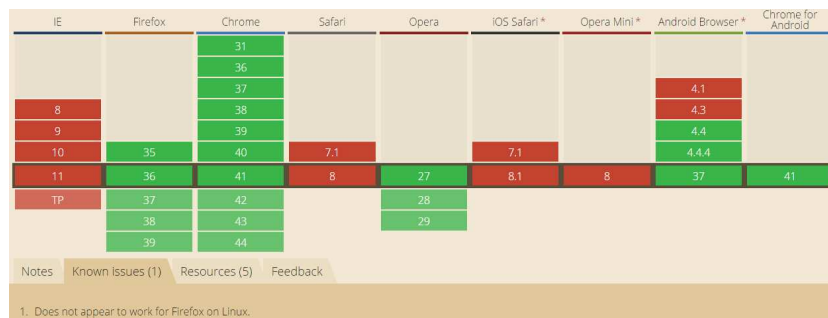


Abbildung 3.15.: Support vom Download-Feld der einzelnen Browser

auch im Firefox, jedoch nicht in Firefox auf Linux-Rechnern funktioniert, was unter "Known issues" aufgeführt ist. Auch der häufig in grossen Unternehmen als vorinstallierter Standard-Browser, Internet Explorer, bietet noch keinen Support für dieses HTML5-Feld an [Deveria, 2015]. Aus diesem Grund ist das Nutzen einer Fallback-Methode unabdingbar. Als Alternative für Browser, welche das HTML5-Attribut `Download` noch nicht zur Verfügung stellen, werden die Daten mit dem MIME-Type `application/octet-stream` heruntergeladen. Da in diesem Fall jedoch die Daten an ein `data`-Attribut angehängt werden und Internet Explorer keine URLs mit mehr als 2048 Zeichen erlaubt, muss auch hier wiederum eine Alternative gefunden werden [Microsoft, 2007]. Der Code, um einen LICARA nanoSCAN als text-Dokument in allen Browsern heruntergeladen zu können, kann im Anhang B entnommen werden.

3.4.2. Architektur

Die Architektur von LICARA NanoSCNA wurde modular aufgebaut, sodass einfach und strukturiert neue Funktionalitäten implementiert werden können. Die Webapp setzt sich aus folgenden Modulen zusammen.

Übersicht der Controller

decisionsupport ist für die Generierung der Daten in der Box Decision Support zuständig. Hierfür holt das Modul Daten aus dem DecisionService und berechnet die Fehlerwerte für die Barcharts und die Netto-Durchschnitte aller Risiko- & Benefitboxen.

3.4. Implementierung

menu ist für die korrekte Weiterleitung bei Klick auf ein Menüitem zuständig. Da dieses Modul ganz zu Beginn beim Initialisieren aufgerufen wird, führt dies auch den Browsersupport-Check durch.

modalinstance ist für die Verarbeitung der eingegeben Daten im Rename-Modal zuständig. Lädt ein User einen zuvor heruntergeladenen Scan hoch, welcher mit demselben Namen schon existiert, hat der User die Möglichkeit, den neuen umzubenennen. Dieses Modul verarbeitet diese Eingaben und leitet diese seinem aufrufenden Controller 'oldscans' weiter.

oldscans ist für die Generierung der Daten für die View 'Your nanoSCANS' zuständig. Zudem besitzt dieses Modul die Funktionen, welche sich um den Upload und Download sowie auch das Löschen von Scans kümmert.

questions ist das Herz der Applikation, da dieses Modul für den Fragebogen zuständig ist. Das json-Dokument, welches alle Fragen für den LICARA-Fragebogen bereit hält, wird hier eingelesen und für die Darstellung aufbereitet. Zudem verfügt dieses Modul über eine `watch`-Funktion, welche auf Veränderungen am Modell hört. Dieses enthält die zur Darstellung benötigten Daten. Beantwortet der User eine Frage durch die Auswahl eines Dropdown-Eintrages, verändert sich dadurch das Modell und das Modul kalkuliert die neuen normalisierten Werte dieser Box und übergibt diese Daten dem Service `decisionService`.

startquestionnaire ist für das Starten des Questionnaires verantwortlich.

Übersicht der Directives HTML und zahlreiche Frameworks bauen ihre Anzahl an Directives (z.B. `input`, `dropdown`, `div`, `span` etc) ständig aus. Um zu vermeiden, dass sich ein für LICARA kreiertes Directive nicht zukünftig mit einem durch z.B. HTML7 eingeführten Directive, überschneidet, wurde jedem Directive das Präfix `lic-`, kurz für LICARA, angehängt.

lic-benbarchart ist für das Generieren der Benefits-Barcharts zuständig. Eine `watch`-Funktion hört, wie im `questions`-Controller, auf Änderungen am Daten-Modell und kalkuliert und aktualisiert dementsprechend seine Werte.

lic-boxestabs ist für die Tab-Darstellung des Questionnaires zuständig. Wie viele Boxen dargestellt werden, hängt von der Anzahl angegebenen Boxen im JSON-Dokument ab. Wird hier eine Box hinzugefügt, werden die Tabs im Menu automatisch um diese Box erweitert.

lic-coordinatesystem ist für das Generieren des Koordinatensystems und seiner Datenpunkte in der 'Decision Support' Box zuständig.

3. Hauptteil

lic-footer ist für die Erstellung des Footers zuständig.

lic-header ist für die Erstellung des Headers zuständig. Zusätzlich hört es auf die Klicks der Language-Buttons, um die Sprache zu ändern.

lic-menu ist für die Erstellung des Menus zuständig.

lic-questwarning ist für die Erstellung eines Warning-Alerts im Questionnaire zuständig. Dies wird zum Beispiel angezeigt, wenn der User, abhängig von seinen Antworten in der Box 'Nanoproduct', darauf hingewiesen soll, dass LICARA nanoSCAN nicht für seine Zwecke geeignet ist, da sein Nanoprodukt nicht als Nano eingestuft wird.

lic-questaccordion ist für die Erstellung der Akkordeons im gesamten Questionnaire zuständig.

lic-risbarchart ist für das Generieren der Risiko-Barcharts zuständig. Eine watch-Funktion hört wie im questions-Controller auf Änderungen am Daten-Modell und kalkuliert und aktualisiert dementsprechend seine Werte.

lic-stoffen-worstcase ist für die Berechnung und Darstellung der Worst-Case-Daten in der Box 'Occupational Health Risk' zuständig.

lic-stoffenmanager ist für die Generierung und das Hören von Klicks der Stoffenmanager-Matrix zuständig.

lic-upload ist für den Upload von LICARA nanoSCAN Text-Files zuständig. Mittels FileReader API wird das Dokument eingelesen und weiterverarbeitet.

Übersicht der Services

decisionService ist für die temporäre Zwischenspeicherung der Daten des Questionnaires zuständig. Beantwortet der User eine Frage, wird dieser Wert an den decisionService weitergeleitet. Wechselt der User von einer Box zur anderen, werden die Daten aus diesem Modul geholt und dargestellt.

languageService ist für die Speicherung der aktuell gewählten Sprache im LocalStorage zuständig.

notificationService ist für das Generieren von Benachrichtigungen zuständig, um dem User ein rasches Feedback auf seine Interaktionen zu geben.

storageService ist für die permanente Speicherung von LICARA nanoSCANs in der IndexedDB zuständig.

3.4.3. Datenmodell

Als Datentyp wurde aufgrund seiner einfachen Lesbarkeit und simplen Regeln JSON gewählt [Nurseitov et al., 2009]. JSON ist ein leichtgewichtiges, textbasiertes und sprachenunabhängiges Datenaustauschformat, welches von der ECMAScript Programmiersprache abstammt. Ein JSON-Dokument besteht aus mehreren Objekten, wobei jedes Objekt von zwei geschwungenen Klammern umschlossen ist und null oder mehr Name/Value Paare enthält. Ein Name ist ein einfacher String, wobei ein Value folgende Werte annehmen kann: Array, Zahl, String, Boolean, null oder ein Objekt [International, 2013]. Die Grundstruktur in unserem JSON sieht wie folgt aus:

```
{
    "general" : {...},
    "menu": {...},
    "boxes": {...},
    "notification" : {...}
}
```

general In **general** sind alle Übersetzungen z.B. für Buttons gespeichert. Bei allen Paaren in diesem Objekt ist die Struktur String/String:

```
"general": {
    "welcome": "Welcome",
    "introduction": "Introduction",
    "yourScansMenu": "Your nanoSCANs",
    "questionnaire": "Questionnaire",
    "box": "Box"
}
```

menu Im Objekt **menu** ist für jede Box ein Menuobjekt erstellt worden. Wünscht man nun dem Questionnaire eine weitere Box hinzuzufügen, kann hier ein neues Objekt angehängt werden. Dadurch wird in der Questionnaire-View ein neuer Tab für die neue Box hinzugefügt.

```
"menu": {
    "-1": {
        "title": "Start",
        "link": "start"
    },
    "0": {
        "title": "Nano product",
        "link": "nanoproduct"
    }
}
```

3. Hauptteil

```
    },  
    "1": {  
        "title": "Environmental benefits",  
        "link": "envben"  
    },  
    "2": {  
        "title": "Economic benefits",  
        "link": "ecoben"  
    },  
    },  
    {...}  
}
```

boxes Im Objekt **boxes** sind die einzelnen Fragen für alle Boxen gespeichert. Die Abbildung 3.16 zeigt am Beispiel der Economic Benefits-Box die Felder, welche in den jeweiligen Boxen vorkommen.

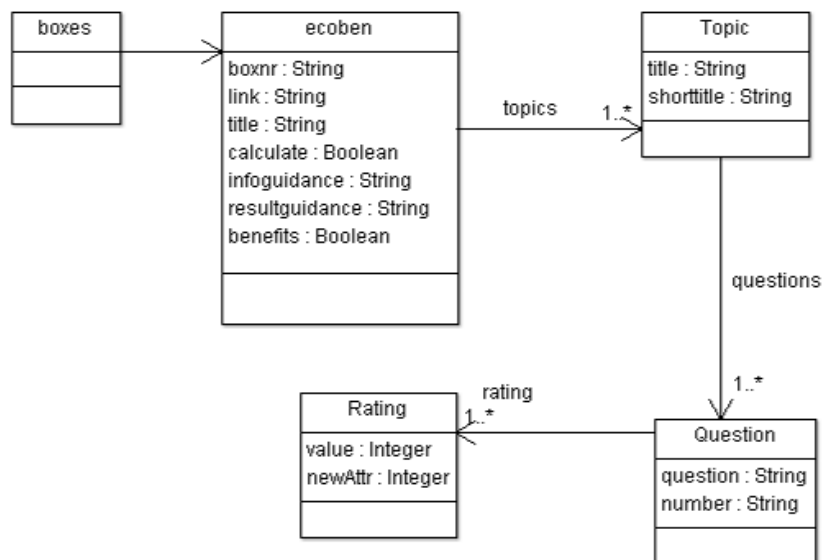


Abbildung 3.16.: Klassendiagramm des JSON-Objektes boxes

notification In diesem Objekt sind alle Übersetzungen für die Benachrichtigungen gespeichert, wie beispielsweise wenn der nanoSCAN erfolgreich gespeichert/gelöscht/hochgeladen wurde.

3.4.4. Kommunikation

Im folgenden Abschnitt wird anhand von zwei Beispielen auf die Kommunikation zwischen den verschiedenen Modulen und AngularJS Komponenten eingegangen.

Auswahl einer Antwort im Dropdown-Menü Die Abbildung 3.17 zeigt auf, was nach der Auswahl einer Antwort im Dropdown-Menü geschieht. Der User befindet sich zum Beispiel in der Ansicht der Box 3. Wählt er hier für die erste Frage eine Antwort im Dropdown-Menü aus, wird diese direkt im Modell, welches die Verbindung zwischen View und Controller ist, gespeichert. Der Controller, in diesem Fall der QuestionController, hört auf Veränderungen am Modell und kalkuliert daraufhin die neuen normalisierten Werte der Box. Die berechneten Werte speichert er im Modell, welches wiederum diese Updates an die View weitergibt. Die View stellt die neuen Werte im UI dar, zum Beispiel im Balkendiagramm unterhalb der Fragen. Gleichzeitig werden die neuen Werte im decisionService zwischengespeichert, um bei einem späteren Wiederaufruf dieser Box die Antworten in den Dropdowns vorausgewählt anzuzeigen.

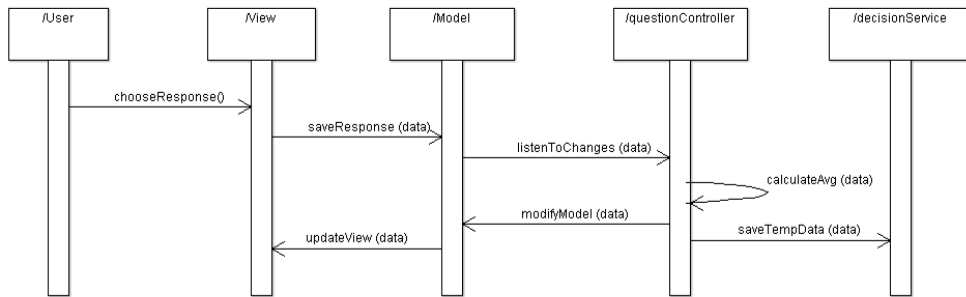


Abbildung 3.17.: Kommunikation beim Auswählen einer Antwort im Dropdown-Menü

Speichern eines Questionnaires Die Abbildung 3.18 zeigt auf, wie und welche Module zusammen interagieren, wenn ein User einen Questionnaire speichern möchte. Der User befindet sich in der 'Decision Support' Box. Er gibt für den Questionnaire einen Namen ein und wählt den Button 'Speichern'. Dieser Button ruft die Funktion `saveQuestionnaire` im DecisionSupportController auf und übergibt den ausgewählten Namen. Der Controller ruft im StorageService eine Funktion auf, welche kontrolliert, ob ein Scan mit dem definierten Namen bereits existiert. Falls ja, wird der User mit einer Benachrichtigung darauf hingewiesen. Falls nein wird der Scan gespeichert und der User auf die 'Your nanoSCANS'-Ansicht weitergeleitet.

3.4.5. Benutzeroberfläche

Introduction Öffnet der User die Webseite von LICARA nanoSCAN, landet er auf der Introduction-Seite. Hier findet er dieselben Informationen, die zuvor in der excelbasierten Version im Introduction-Tab auffindbar waren. Durch die Einführung der Akkordeons können Informationen zugeklappt werden, um die Übersichtlichkeit der angezeigten Inhalte zu gewährleisten. Am oberen Rand neben dem Logo befinden sich die Sprachicons. Klickt der User auf ein Icon,

3. Hauptteil

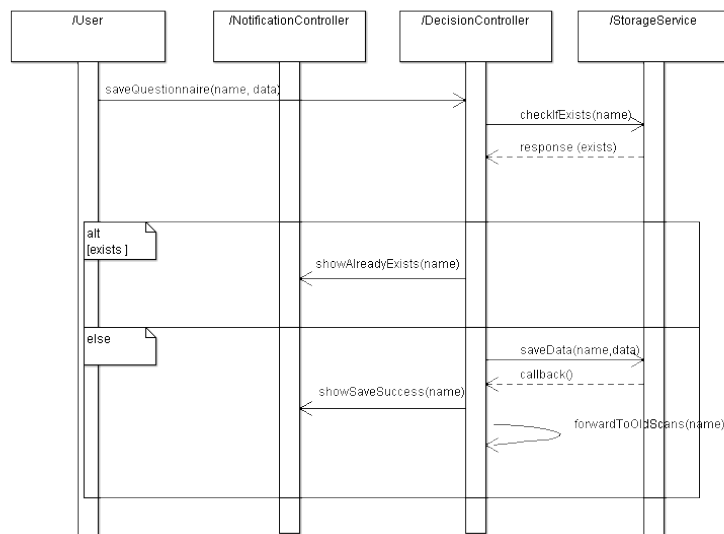


Abbildung 3.18.: Kommunikation beim Speichern eines Questionnaires

wird die Sprache auf der gesamten Webseite ohne Pagereload gewechselt.

Your nanoSCANS Am linken Rand befindet sich die Navigation. Der User kann hier zwischen Introduction, Your nanoSCANS und Start nanoSCAN auswählen. Beim Klick auf 'Your nanoSCANS' wird der User auf eine neue Seite geführt. Dort werden die bereits ausgefüllten und gespeicherten nanoSCANS angezeigt. Übersichtshalber wurde auch hier auf die Akkordeon-Ansicht zurückgegriffen. Standardmässig werden hier nur der gewählte Name des nanoSCANSs, das Erstell- und wenn vorhanden das Editier-Datum angezeigt. Klickt der User auf den Namen, öffnet sich die Detailansicht des gewählten nanoSCANSs. Die Detailansicht liefert die Barcharts und die Entscheidungshilfe sowie die Möglichkeit den nanoSCAN als PDF oder als Text herunterzuladen sowie zu löschen und zu editieren.

Editieren Wählt der User die Funktion editieren, wird er auf den Navigationspunkt Start nanoSCAN weitergeleitet und die entsprechenden Fragen werden mit den zuvor gewählten Antworten ausgefüllt. Ein Benachrichtigungs-Popup im oberen rechten Bereich der Webseite weist den Benutzer darauf hin, dass er nun diesen Scan bearbeitet.

Löschen Wird vom Benutzer diese Funktion ausgewählt, wird der gewünschte nanoSCAN unwiderruflich gelöscht. Über den Erfolg des Löschen des nanoSCANSs wird oben rechts durch eine Benachrichtigung informiert.

Herunterladen Wählt der Benutzer die Funktion 'Herunterladen als Text' oder 'als PDF', wird im Hintergrund das Dokument für das Herunterladen generiert und bereitgestellt. Da dieser Prozess, vor allem für die PDF-Version einen Moment dauert und dadurch je nach Browser diesen eventuell kurz einfriert, wird der Benutzer über eine Benachrichtigung darüber informiert. Sobald das Dokument heruntergeladen worden ist, befindet sich dies im Standardordner für Downloads aus dem Browser.

Hochladen Unterhalb der verfügbaren nanoSCANS befindet sich der Uploadbereich. Hier kann der User ein zuvor als Text-Dokument heruntergeladenen nanoSCAN hochladen. Wählt der Benutzer einen falschen Dateityp oder ein ungültiges Dokument zum Upload aus, wird dieser auf den Fehler hingewiesen. Wählt er ein Dokument aus, dessen nanoSCAN-Name bereits existiert, wird ein Modal geöffnet, in dem der User für den gewünschten nanoSCAN einen neuen Namen eingeben oder den Upload abbrechen kann. War der Upload erfolgreich, wird dies durch eine Benachrichtigung signalisiert und der User wird auf die Akkordeon-Ansicht des hochgeladenen nanoSCAN weitergeleitet.

Start nanoSCAN Wählt der User den Navigationspunkt Start nanoSCAN führt ihn dies zum Herzstück der Applikation, zum LICARA nanoSCAN Fragebogen. Am oberen Rand befindet sich nun die Tab-Navigation, in dem per Klick zwischen den einzelnen Boxen gewechselt werden kann. Die Farben der Boxen entsprechen denjenigen der excelbasierten Version. Der User kann zwischen den einzelnen Boxen hin- und herswitchen, ohne dass die Daten verloren gehen oder er die Informationen selber zwischenspeichern muss. Unterhalb der Fragen befinden sich links der Zurück- und Rechts der Vorwärts-Button, um die Navigation zwischen den Boxen für den Benutzer zu erleichtern. Zudem befindet sich am Ende des Inhaltes ein 'Top'-Button, um hoch zur Navigation zu scrollen. Der Resultate-Bereich zeigt den normalisierten Wert der Fragen in der geöffneten Box an. Gleich darunter befindet sich der Barchart, welcher farblich die Werte der einzelnen Topics darstellt. Diese Balken verändern je nach Antwort dynamisch die Farbe und die Breite, um dem User sofortiges Feedback zu geben, welche Auswirkungen die gewählte Antwort auf das Schlussresultat hat. In der Box 'Occupational Risk' sollen die Lösungen des Stoffenmanager-Tools eingetragen werden. Als Eintragungsmöglichkeit wurde hier die Matrixform gewählt, da diese Bedienung für den Benutzer intuitiver ist. Fährt er mit der Maus über ein Feld in der Matrix, erfährt er mit Hilfe eines Tooltips die daraus resultierenden Werte, was eine für den Kunden transparente Berechnung der Boxwerte ermöglicht.

Decision Support Wählt der Benutzer die siebte und letzte Box, gelangt er zur 'Decision Support' Box. Ganz oben befindet sich der Speichern-Bereich. Hier kann der User einen Namen für den Questionnaire eingeben, unter welchem dieser gespeichert werden soll. Nach erfolgreichem Speichern wird der Benutzer zur 'Your nanoSCANS'-Seite weitergeleitet. Zudem hat der Benutzer

3. Hauptteil

hier die Möglichkeit, alle eingegebenen Daten zu löschen, um den Questionnaire auf Null zurückzusetzen. Gleich unterhalb dieses Bereichs werden die Risiko- und Benefits-Boxen gegenübergestellt und die einzelnen normalisierten Werte grafisch in einem Balkendiagramm dargestellt. Fährt der User über die Balken, sieht er die die Werte für die Topics pro Box. Auch bei den Fehlerwerten, die durch das Leerlassen einzelner Fragen berechnet werden, können im Gegensatz zur excelbasierten Version weitere Informationen per Mouseover abgelesen werden. Da in dieser Box in der excelbasierten Version das Problem bestand, dass zu viel Text die Übersichtlichkeit nicht gewährleistet, wurden hier für die einzelnen Informationen wie in den anderen Boxen ebenfalls die Akkordeons gewählt. Dadurch kann per Klick auf ein Akkordeon Text angezeigt oder ausgeblendet werden.

3.5. Evaluierung

Zur Evaluierung wurde anfangs April eine erste Version Mitarbeitern der EM-PA und der TNO zum Testen freigegeben. Das daraus resultierende Feedback wurde analysiert und bei einem Re-Design umgesetzt.

3.5.1. Feedback

Generell waren die Tester sehr zufrieden mit der Webapplikation, welche einen sehr professionellen Eindruck hinterliess. Die Akkordeon-ähnliche Darstellung von Inhalten wurde als eine sehr elegante Lösung wahrgenommen. Zudem wurden die dynamischen Balkendiagramme als sehr nützlich eingestuft. Auch die Speicherfunktion wurde gegenüber der excelbasierten Version als sehr aufwertend empfunden. Wie die Tabelle 3.2 zeigt, wurden die Rückmeldungen und Verbesserungsvorschläge in die Kategorien Design, Content und Funktionalität unterteilt.

3.5.2. Umsetzung

Die Änderungsvorschläge für das Design und den Inhalt wurden komplett umgesetzt. Die Funktionalitäten hingegen wurden nicht gemäss den Änderungswünschen angepasst.

Die Funktion, ein nanoSCAN herunterzuladen, wurde nicht wie von einer Testperson gewünscht, deaktiviert. Das heruntergeladene Dokument hat zwar keinen direkt ersichtlichen Mehrnutzen, es kann jedoch per Mail weitergeleitet werden. Der Empfänger hat die Möglichkeit, dieses Dokument in der Webapplikation hochzuladen und den nanoSCAN im Detail zu betrachten und gegebenenfalls zu bearbeiten. Diese Verhaltensweise wird ebenfalls bei der Precautionary Matrix vom BAG angewendet, welches von der betreuenden Person als sehr nützlich empfunden wurde.

Die zweite und letzte Rückmeldung, welche nicht umgesetzt wurde, war das Zwischenspeichern, falls ein User versehentlich die Seite verlässt. Das Speichern des nanoSCANS nach jedem Beantworten einer Frage würde zu grossen Performanz-Einbussen führen. Zudem kann ein nanoSCAN zu jeder Zeit in der letzten Box gespeichert werden, um zu einem späteren Zeitpunkt weiterzufahren. Der User wird in der Start-Box neu explizit darauf hingewiesen.

3. Hauptteil

Design	Content	Funktionalität
In der Box 5 soll die boxähnliche Matrixdarstellung durch ein Hintergrundbild ersetzt werden, um keine klaren, farblichen Grenzen zwischen den einzelnen Werte zu kreieren.	Ein Bild auf der Introduction-Seite soll entfernt, die restlichen mit einem Label beschriftet werden.	Wenn möglich sollen die Werte vom Questionnaire zwischengespeichert werden, damit der User, falls dieser versehentlich den Tab schliesst oder eine neue Seite aufruft, beim Wiederaufruf der Seite dort weiterarbeiten kann, wo er aufgehört hat.
In der Box 5 soll ein Hinweis hinzugefügt werden, welche Werte vom Stoffenmanager in die Webapplikation eingefügt werden sollen.	Bei der Frage 0.10 soll der Empfehlungstext abgeändert werden.	Das Text-Dokument, welches heruntergeladen werden kann, hat keinen Zusatznutzen und soll daher nicht zum Download zur Verfügung stehen.
Bei den Risiko-Barcharts war der Balken bei einem Wert von > 0.5 rot eingefärbt. Dieser soll jedoch erst ab > 0.67 in rot erscheinen.	Das EU-Logo soll integriert werden.	
Der Background im PDF soll weiss bleiben.		
Das LICARA-Logo soll ins PDF integriert werden.		

Tabelle 3.2.: Übersicht der Rückmeldungen nach der Testphase

3.6. Schwierigkeiten

Während der Erstellung von LICARA nanoSCAN sind verschiedene Probleme aufgetaucht, auf welche in den nächsten Abschnitten eingegangen wird.

3.6.1. Cross-Browser-Kompatibilität

Wie bereits im Abschnitt 3.4.1 beschrieben, ist die Cross-Browser-Kompatibilität sehr wichtig, um möglichst vielen Nutzern die Möglichkeit zu geben, die Webapplikation zu nutzen. Bei zwei Funktionalitäten erwies sich dies jedoch als schwierig.

Bei der Upload-Funktionalität wird auf die FileReader API zugegriffen, um das ausgewählte Dokument aus dem Filesystem hochzuladen und einzulesen. Jedoch wird diese API zwar von allen Browsern in der aktuellen Version unterstützt, ältere Versionen liefern diesen Support jedoch nicht mit. Dasselbe Problem taucht beim Speichertyp IndexedDB auf. Das Gute daran: Browser, welche die FileReader API nicht unterstützen, verfügen auch nicht über diese IndexedDB-Speichermethode. Aus diesem Grund würde ein File-Upload auch kein Nutzen liefern, da das Dokument zwar eingelesen, jedoch nicht gespeichert und dadurch für den User nicht dargestellt werden kann. Benutzt der User eine ältere Version, welche einer dieser beiden Technologien nicht unterstützt, wird er mit einer Benachrichtigung darauf hingewiesen, einen aktuelleren Browser zu benutzen, um von allen Funktionalitäten Gebrauch machen zu können.

3.6.2. Analyse der Box-Kalkulationen

Nach Beantworten einer Frage werden in jeder Box im Hintergrund verschiedene Kalkulationen durchgeführt, um die normalisierten Werte zu berechnen. Die Excel-Version, welche zur Verfügung stand, war durch einen Code geschützt. Das Ausfüllen des Fragebogens war zwar möglich, die Ansicht jedoch, in welcher die Berechnungen stattfinden, blieb jedoch verwehrt. Aus diesem Grund galt es zuerst zu verstehen, welche Antwort zu welcher Frage welchen Wert hat und wie sich dies auf das Endresultat auswirkt. Durch ein langwieriges Trial-And-Error-Vorgehen konnten so die entsprechenden Werte herausgefunden werden. Um diese Werte auch kontrollieren zu können, wurden drei bereits ausgefüllte nanoSCANs bereitgestellt, welche als Referenz benutzen werden konnten.

3.6.3. Balkendiagramme im PDF

Bei der Generierung des PDF's ist man auf die Grenzen des Frameworks gestoßen, welches bei der Erstellung des PDF's helfen sollte. Die beiden Bildformate png und jpeg werden vom Open-Source-Framework 'pdfmake' unterstützt, um ins PDF eingefügt zu werden. Das Format svg, welches für die dynamische Generierung der Balkendiagramme benutzt wird, wird jedoch nicht unterstützt. Deswegen wird das svg zuerst mittels eines Canvas in ein png umgewandelt,

3. Hauptteil

welches in einem zweiten Schritt ins PDF inkludiert werden sollte. Dies war der Zeitpunkt, an dem ein Bug im Framework entdeckt wurde: Das ausgewählte Bild wird nur als ein dicker, schwarzer Balken dargestellt. Das daraufhin erstellte Issue wurde vom Entwickler sofort als Bug eingestuft, jedoch bis zum Ende dieser Arbeit noch nicht gefixt. Ein gleichwertiges Framework mit ähnlichem Funktionsumfang konnte in der Open-Source-Welt nicht gefunden werden. Um die Benutzer darauf aufmerksam zu machen, dass die Resultate im PDF nicht dargestellt werden, erscheint vor dem Download der Hinweis, ein Screenshots zu erstellen, falls diese lokal verfügbar sein sollen. Das dazugehörige Modal zeigt die Abbildung 3.19.

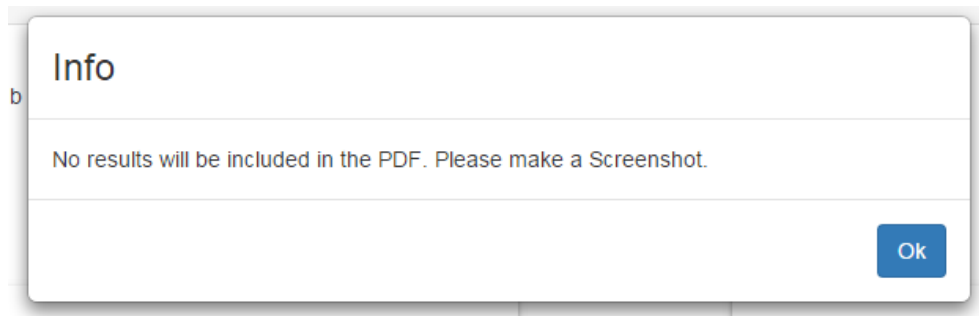


Abbildung 3.19.: Hinweis auf fehlende Balkendiagramme

4. Fazit und Ausblick

4.1. Fazit

Die EMPA hat in Zusammenarbeit mit dem Unternehmen TNO ein Excel-basiertes Decision Support Tool entwickelt, welches Herstellern von innovativen Materialien bei der Entscheidung helfen soll, ob die Verwendung von ausgewählten Nanopartikel in der Herstellungsphase sinnvoll ist.

Das Ziel der Arbeit war es, diese Excel-Version ins Web zu portieren, um die Benutzerfreundlichkeit sowie die Nutzungshäufigkeit zu erhöhen. Die kritische Evaluation hat gezeigt, dass das Design sowie die neuen Funktionen der Webapplikationen überzeugten. Vor allem die Speicherfunktion von nanoSCANS verleiht der webbasierten Version einen grossen, zusätzlichen Nutzen. Für ältere Browser steht die Speicher sowie die Upload-Funktionalität jedoch nicht zur Verfügung. Das manuelle Hinzufügen, Löschen und Editieren von Fragen, Antworten und Boxen kann von Mitarbeitern der EMPA und TNO selbstständig durchgeführt werden. Dies führt dazu, dass der Bedarf eines Administrators und die dadurch entstandene Abhängigkeit wegfällt.

4.2. Ausblick

Sobald der Fehler im Framework 'pdfmake' behoben ist, wird die Weiterentwicklung des Generierens des PDF's aufgenommen, um dem User alle relevanten Inhalte in einem PDF zur Verfügung zu stellen.

Die Webapplikation wird in Zukunft auf den Webseiten der EMPA und der TNO zur Benutzung zur Verfügung stehen.

Durch den modularen Aufbau der Webapplikation steht der deutschen Version, welche zur Zeit in Planung steht, nichts mehr im Wege.

Literaturverzeichnis

- [BAG, 2013] BAG (2013). Vorsorgeraster synthetische nanomaterialien. <http://www.bag.admin.ch/nanotechnologie/12171/12174/index.html?lang=de>. Zuletzt aufgerufen am 8. März 2015.
- [Bostock, 2013] Bostock, M. (2013). D3.js - data-driven documents. <http://d3js.org/>. Zuletzt aufgerufen am 25. März 2015.
- [Bostock et al., 2011] Bostock, M., Ogievetsky, V., and Heer, J. (2011). D3: Data-driven documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*.
- [Chaniotis et al., 2014] Chaniotis, I., Kyriakou, K.-I., and Tselikas, N. (2014). Is node.js a viable option for building modern web applications? a performance evaluation study. *Computing*, pages 1–22.
- [Choudhary et al., 2010] Choudhary, S. R., Versee, H., and Orso, A. (2010). Webdiff: Automated identification of cross-browser issues in web applications. In *ICSM*, pages 1–10. IEEE Computer Society.
- [Christensson, 2009] Christensson, P. (2009). Cross-browser. <http://techterms.com/definition/crossbrowser>. Zuletzt aufgerufen am 01. April 2015.
- [Deveria, 2015] Deveria, A. (2015). Download attribute. <http://caniuse.com/#search=download>. Zuletzt aufgerufen am 01. April 2015.
- [Empa, 2014] Empa, TNO, N. (2014). Licara guidelines for the sustainable competitiveness of nanoproducts.
- [Freeman et al., 2004] Freeman, E., Freeman, E., Bates, B., and Sierra, K. (2004). *Head First Design Patterns*. O’ Reilly & Associates, Inc.
- [Google, 2015] Google (2015). Angularjs, html enhanced for web apps. <https://angularjs.org/>. Zuletzt aufgerufen am 25. März 2015.
- [Hansen et al., 2013] Hansen, S., Jensen, K., and Baun, A. (2013). Nanoriskcat: a conceptual tool for categorization and communication of exposure potentials and hazards of nanomaterials in consumer products. *Journal of Nanoparticle Research*, 16(1).

Literaturverzeichnis

- [International, 2013] International, E. (2013). The json data interchange format. <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>. Zuletzt aufgerufen am 10. April 2015.
- [Mahemoff, 2010] Mahemoff, M. (2010). Client-side storage. <http://www.html5rocks.com/en/tutorials/offline/storage/>. Zuletzt aufgerufen am 16. März 2015.
- [Malsch et al., 2015] Malsch, I., Subramanian, V., Semenzin, E., Hristozov, D., and Marcomini, A. (2015). Supporting decision-making for sustainable nanotechnology. *Environment Systems and Decisions*, 35(1):54–75.
- [Mark Otto, 2015] Mark Otto, J. T. (2015). Bootstrap, the world’s most popular mobile-first and responsive front-end framework. <http://getbootstrap.com/>. Zuletzt aufgerufen am 25. März 2015.
- [McNeil, 2014] McNeil, P. (2014). Taking donald norman’s design principles to web design. <http://www.howdesign.com/web-design-resources-technology/donald-normans-design-principles-modern-web-design/>. Zuletzt aufgerufen am 7. April 2015.
- [Microsoft, 2007] Microsoft (2007). Maximum url length is 2,083 characters in internet explorer. <https://support.microsoft.com/en-us/kb/208427/>. Zuletzt aufgerufen am 8. Mai 2015.
- [Nielsen, 2001] Nielsen, J. (2001). 113 design guidelines for homepage usability. <http://www.nngroup.com/articles/113-design-guidelines-homepage-usability/>. Zuletzt aufgerufen am 24. April 2015.
- [Nikunj Mehta, 2015] Nikunj Mehta, Jonas Sicking, E. G. A. P. J. O. J. B. (2015). Indexed database api. <http://www.w3.org/TR/IndexedDB>. Zuletzt aufgerufen am 01. April 2015.
- [Norman, 1988] Norman, D. (1988). *The Design of Everyday Things*. Number Nr. 842 in *The Design of Everyday Things*. Doubleday.
- [Nurseitov et al., 2009] Nurseitov, N., Paulson, M., Reynolds, R., and Izurieta, C. (2009). Comparison of JSON and XML Data Interchange Formats: A Case Study. pages 157–162. CAINE.
- [Pillora, 2015] Pillora, J. (2015). Notify.js - a simple, versatile notification library. <http://notifyjs.com/>. Zuletzt aufgerufen am 25. März 2015.
- [Powell, 2012] Powell, A. (2012). db.js. <http://aaronpowell.github.io/db.js/>. Zuletzt aufgerufen am 01. April 2015.
- [van Kronenburg 14, 2015] van Kronenburg 14, L. (2015). Stoffenmanager. <https://stoffenmanager.nl>. Zuletzt aufgerufen am 17. April 2015.

A. Code für die Darstellung der Balkendiagramme bei unterschiedlichen Bildschirmgrößen

```
<div class="col-lg-5 col-xs-12">
  <div class="topic">
    <h4 class="chart-header" translate>
      general.benefits</h4>
    <lic-benbarchart data="data.bboxes"></
      lic-benbarchart >
  </div>
</div>
<div class="col-lg-5 col-lg-offset-1 col-xs-12">
  <div class="topic">
    <h4 class="chart-header" translate>
      general.risks</h4>
    <lic-risbarchart data="data.bboxes"></
      lic-risbarchart >
  </div>
</div>
```


B. Code zum Download als Text-File

```
var strMimeType = 'application/octet-stream;charset=utf-8';
var fileName = item.nameOfQuestionnaire;

// IE10-
if (this.isIEBelow10()) {
    var frame = document.createElement('iframe');
    document.body.appendChild(frame);
    frame.contentWindow.document.open("text/html",
        "replace");
    frame.contentWindow.document.write('sep=\r\n'
        + item);
    frame.contentWindow.document.close();
    frame.contentWindow.focus();
    frame.contentWindow.document.execCommand('
        SaveAs', true, fileName);
    document.body.removeChild(frame);
    return true;
}

// IE10+
if (navigator.msSaveBlob) {
    return navigator.msSaveBlob(new Blob([JSON.
        stringify(item, function (key, val) {
            return val;
        })], {
        type: strMimeType
    }), fileName + '.txt');
}
else {
    //all other browsers except IE
    var data = "data:" + strMimeType + "," +
        encodeURIComponent(JSON.stringify(item));
    window.open(data);
}
}
```