




**Maßnahmenplan Barrierefreiheit zum Statusdokument\_Shopfloor**

Anwendung:	Shopfloor Webapp
Version der Anwendung:	1.0
Version des Maßnahmenplans:	1.0

Nr.	Problembeschreibung	S	B	M	H	K	Beispielhafter Lösungsvorschlag	Entwicklungsmaßnahme	Kommentar	Hinweise
[1]	Fehlermeldungen sind nicht wahrnehmbar	☹☹☹	☹☹☹	-	-	-	- Nachdem das System einen Fehler festgestellt hat, sollte eine Fehlermeldung in Textform angezeigt werden (z.B. beim Verlassen des Feldes oder beim Absenden des Formulars). - An der Fehlermeldung sollten das betroffene Feld sowie die Fehlerursache erkennbar sein. - Die Fehlermeldungen über dem Formular sollten das fehlerhafte Feld konkret benennen. - Die Fehlermeldungen sollten mindestens 10 Sekunden lang sichtbar sein.	- Meldungen 10 Sekunden sichtbar		
[2]	Statusmeldungen sind nicht wahrnehmbar	-	☹☹☹	☹☹☹	-	-	- Statusmeldungen sollten von blinden Nutzern direkt wahrgenommen werden können, indem sie mit der ARIA-Rolle role="alert" ausgezeichnet werden. - Statusmeldungen sollten mindestens 10 Sekunden lang sichtbar sein. - Meldungen sollten sich nicht überlagern.	- role="alert" hinzugefügt => getestet mit ChromeVox - Meldungen 10 Sekunden sichtbar - Alte Meldung wird beim erscheinen einer neuen Meldung wieder ausgeblendet.		
[3]	Einige Buttons zum Ausblenden besitzen keine Beschriftung	-	☹☹	-	-	-	- Für die Buttons ohne sichtbare Beschriftung, wie z.B. Icon-Buttons, sollte das Attribut aria-label verwendet werden, um die Aktion eindeutig zu beschreiben: - <button aria-label="Aufklappen" onclick="myDialog.close()">...</button>	- aria-label bei allen icon-only buttons hinzugefügt		
[4]	Startseite-Link ist nicht aussagekräftig beschriftet	-	☹	-	-	-	- Die Alternativtexte der Grafiken sollten aussagekräftig gestaltet werden, z.B. „Shopfloor Startseite“.	- Logo mit "Shopfloor Startseite" beschriftet und Link hinterlegt - Footer Logo gibt es nicht mehr		
[5]	Viele Buttons besitzen englische Beschriftungen	-	☹	-	-	-	- In einer deutschsprachigen Anwendung sollten englische Beschriftungen und Begriffe vermieden werden.	- close-button-label="Schließen" zu Modal-Tag hinzugefügt		
[6]	Icon-Buttons sind inkorrekt beschriftet	-	☹	-	-	-	- Für die Buttons ohne sichtbare Beschriftung, wie z.B. Icon-Buttons, sollte das Attribut aria-label verwendet werden, um die Aktion eindeutig zu beschreiben: - <button aria-label="Close" onclick="myDialog.close()">...</button>	- Button mit aria-label Beschriftet (mit ChromeVox getestet)		
[7]	Die Auswahl-Elemente sind mit der Tastatur nicht aktivierbar	-	☹☹	☹☹	-	-	- Alle Funktionen sollten mit der Tastatur (Eingabe- oder Leertaste) aktiviert werden können. - Interaktive Elemente sollten mit den HTML-Elementen <a>, <button>, <input>, <textarea> oder <select> umgesetzt werden. Links sollten ein nicht-leeres href-Attribut enthalten.		- deprecated => neu ist eine Checkbox mit 2 (3) Status vorhanden	
[8]	Nicht-interaktive	-	☹	-	-	-	- Nicht-interaktive Elemente sollten nicht als		- bei der im Testprotokoll	
[9]	Hierarchische Überschriften sind nicht korrekt ausgezeichnet	-	☹	-	-	-	- Die Überschriftenelemente <h1> bis <h6> sollten hierarchisch korrekt eingesetzt werden, d. h. Hauptüberschriften erhalten das <h1>-Element, Überschriften der Ebene 2 das <h2>-Element usw.	- "Beginn" und "Ende" auf <h2> geändert	- teilweise deprecated da neues Design	
[10]	Das Pop-up zur Datumsauswahl besitzt englische Überschrift	-	☹	-	-	-	- In deutschsprachiger Anwendung sollten englische Begriffe vermieden werden.		- deprecated => Scale Update bringt Übersetzung	
[11]	Labels sind mit den Eingabefeldern nicht verknüpft	-	☹	☹	-	-	- Beschriftung und Formularfeld sollten im HTML-Markup verknüpft werden. Dazu sollte das for-Attribut des <label>-Elements mit der id des Formularfeldes übereinstimmen. - <label for="vorname">Vorname</label> - <input type="text" id="vorname" .../>			
[12]	Platzhaltertexte werden vor dem zugewiesenen Wert ausgegeben	-	☹	-	-	-	- Beschriftung und Formularfeld sollten im HTML-Markup verknüpft werden. Dazu sollte das for-Attribut des <label>-Elements mit der id des Formularfeldes übereinstimmen. - <label for="vorname">Vorname</label> - <input type="text" id="vorname" .../>			
[13]	Im Hochkontrastmodus sind die Checkboxes nicht als interaktiv erkennbar	☹	-	-	-	-	- Für interaktive Elemente sollten die Standard-HTML-Elemente verwendet werden. Dies gilt insbesondere für Schalter, Links, Kartellie, Menüeinträge, Ausklapplisten und Checkboxes.		- die Checkboxes wurden durch Scale wahrscheinlich angepasst oder ich habe den falschen Kontrast-Modus benutzt	
[14]	Im Hochkontrastmodus sind die Rahmen einiger Elemente nicht sichtbar	☹	-	-	-	-	- Es sollte sichergestellt werden, dass sich mit der Windows-Kontrastanpassung die Kontraste aller Inhalte anpassen lassen. - Die Elemente sollten per CSS einen Rahmen (border) erhalten. Dabei kann mit nicht-sichtbaren Attributen, z.B. einer transparenten Rahmenfarbe gearbeitet werden, da diese dann bei einer Kontrastanpassung sichtbar wird.	@media screen and (-ms-high-contrast: active) { .card { border: 1px solid black; } }  @media (forced-colors: active) { .card { border: 1px solid black; } }		
[15]	Bei der Schriftgrößenanpassung sind einige interaktive Elemente schwer wahrnehmbar	☹	-	-	-	-	- Bei der Vergrößerung von Schrift sollte darauf geachtet werden, dass sich die Elemente der Benutzeroberfläche mitskalieren, um Überdeckungen zu vermeiden. - Abstände zwischen Textblöcken sollten so eingesetzt werden, dass eine Überlagerung der Textblöcke bei Vergrößerung ausgeschlossen wird.		- bei Shopfloorablauf und -lage: deprecated da neues Design - bei Combobox: Ticket INNO-5235 zur Fertigstellung erstellt	
[16]	In der mobilen Ansicht kann das Untermenü nicht geöffnet werden	☹	-	-	-	-	- In der mobilen Ansicht sollten alle Anwendungsinhalte erreichbar und bedienbar sein.		- deprecated da das Mobile Menü mittlerweile hinzugefügt wurde - hat zudem keine Relevanz, da die Anwendung nicht für Mobile Endgeräte gedacht ist	
[17]	Viele Elemente überlappen sich bei der mobilen Ansicht	☹	-	-	-	-	- Die Website sollte so umgesetzt werden, dass diese automatisch an vorgegebene Bildschirm- und Fenstergrößen angepasst wird. Die Inhalte sollten dabei in ein einspaltiges Gestaltungsraster umbrechen. Infolgedessen sollten alle Anwendungsinhalte lesbar und bedienbar sein.	- bei Kennzahl ansehen: class "col-s-12" hinzugefügt	- bei Shopfloorablauf und -lage: deprecated da neues Design - bei Kennzahl ansehen (Combobox): wird in Ticket INNO-5235 fertiggestellt	
[18]	Alle Seiten besitzen den gleichen Titel	☹	☹	-	-	-	- Mit Hilfe des <title>-Elements im Dokumentenheader sollten für alle Seiten unterschiedliche Titel vergeben werden, die eindeutig sind und auf den Inhalt der Seite hinweisen.	- im VueRouter => meta den Object-Key "title" hinzugefügt und für jede Komponente einen speziellen Titel vergeben		
[19]	Seitensprache ist nicht ausgezeichnet	-	☹	-	-	-	- Die vorherrschend verwendete Sprache sollte mit <html lang="de"> gekennzeichnet sein.	- in index.ejs Attribut "lang" mit der value "de" im tag "html" hinzugefügt		
[20]	Pflichtfelder sind nicht ausgezeichnet	☹	☹	☹	-	-	- Pflichtfelder sollten durch einen textlichen Hinweis im Label des Feldes gekennzeichnet sein, z.B. durch einen * (Stern). Zusätzlich können Felder auch farblich markiert werden, um besser darauf aufmerksam zu machen. - Falls die Formularfelder kein <label>-Element besitzen, kann die Pflichtfeldkennzeichnung über das title-Attribut beim Formularfeld eingebunden werden. Alternativ können Pflichtfelder mit dem Attribut aria-required="true" ausgezeichnet werden, damit der Screenreader die Pflichtfelder erkennen und entsprechend ausgeben kann. In diesem Fall ist darauf zu achten, dass die Pflichtfeldkennzeichnung für sehbehinderte Nutzer bei der Kontrastanpassung z.B. über einen breiteren Rahmen zu erkennen ist. Auf die abweichende Auszeichnung der Pflichtfelder sollte in der Hilfe und auf den Seiten selbst hingewiesen werden.	- bei Pflichtfeldern wurde an das Label der * hinzugefügt und an das Input aria-required="true" - bei nicht beschreibbaren Feldern wurde aria-readonly="true" hinzugefügt - aria-label an "Anwendungs-Ebenen" hinzugefügt		
[21]	Einige Syntaxfehler sind vorhanden	-	☹	-	-	-	- Die Seiten sollten keine WCAG-relevanten Syntaxfehler enthalten. - Zur Prüfung der Seiten kann der Nu Html Checker ( <a href="https://validator.w3.org/hu/">https://validator.w3.org/hu/</a> ) verwendet werden. Anschließend können mit dem Parsing Bookmarket ( <a href="https://cdpn.io/stevefdebug/VR2dGJ/">https://cdpn.io/stevefdebug/VR2dGJ/</a> ) die Fehler gefiltert werden, die relevant für die Erfüllung dieses Prüfkriteriums sind.			
[22]	Buttons zum Schließen werden ohne Rolle ausgegeben	-	☹	-	-	-	- Soweit möglich, sollten Standardelemente eingesetzt werden, um die Funktionalitäten bereit zu stellen. - Alternativ sollten die korrekten ARIA-Attribute verwendet werden, damit blinde Nutzer die Rolle und den Status der Elemente wahrnehmen können (siehe <a href="http://www.w3.org/TR/wai-aria-complete">http://www.w3.org/TR/wai-aria-complete</a> , <a href="http://www.w3.org/TR/wai-aria-in-html">http://www.w3.org/TR/wai-aria-in-html</a> ).	- Scale-Attribute "dismiss-text" an tag und "close-button-label" an Modal gesetzt		

## Hinweise

S	Nutzung mit eingeschränktem Sehvermögen oder Nutzung ohne Farbwahrnehmung
B	Nutzung ohne Sehvermögen
M	Nutzung mit eingeschränkter Handhabung oder Kraft, Nutzung mit eingeschränkter Reichweite
H	Nutzung ohne Hörvermögen, Nutzung mit eingeschränktem Hörvermögen oder Nutzung ohne Sprechvermögen
K	Nutzung mit kognitiven Beeinträchtigungen oder Verringerung von Anfallsauslösern bei Photosensibilität
-	nicht relevant für die Benutzergruppe
n. b.	nicht betrachtet
n. a.	nicht anwendbar
n. p.	nicht prüfbar
	Eine Zugänglichkeitsblockade führt dazu, dass eine aufgabenrelevante Funktion nicht bedienbar oder eine aufgabenrelevante Information nicht wahrnehmbar ist.
	Eine Zugänglichkeitshürde führt dazu, dass eine aufgabenrelevante Information schwer verständlich oder schwer wahrnehmbar ist bzw. eine aufgabenrelevante Funktion oder Information nur durch Umgehung des Problems mittels einer Hilfskonstruktion genutzt werden kann.
	Eine leichte Zugänglichkeitseinschränkung führt dazu, dass Informationen oder Funktionen erst durch bestimmte Nutzeraktionen zugänglich werden (z. B. durch das Ändern von Browsereinstellungen, das Anpassen von Einstellungen der assistiven Technologie oder durch Anwendungsschulungen) bzw. der Nutzer gestört oder abgelenkt wird.
✓	Korrekt umgesetzte Anforderungen an die Zugänglichkeit.
schwerwiegend	Der Mangel muss behoben werden, damit die betroffene Benutzergruppe die IKT ohne Fremde Hilfe nutzen kann.
erheblich	Der Mangel muss behoben werden, damit die betroffene Benutzergruppe die IKT ohne besondere Erschwernis nutzen kann.
leicht	Der Mangel muss behoben werden, damit die Benutzergruppe die IKT in der allgemein üblichen Weise nutzen können.
gelöst	Der Mangel wurde behoben.