

RELEASE LETTER

Downtimecapture

04.07.2023

Inhalte der Auslieferung:

Im Repository befinden sich zum einen der Code und zum anderen die Dokumente, die im Laufe des Projektes erstellt wurden.

- “downtimecapture”: Bestandteil ist der Code des Projektes, unterteilt in
 - Backend mit NestJS
 - Speicherung der dtc-Message in eine MySQL-Datenbank
 - Frontend mit Angular
 - Erfassung und Speicherung von downtime Messages
 - Responsive Webdesign
 - Validierung der Eingabefelder
 - Erfolgreiche Statusmeldungen
- “project-docs”: Dieser Ordner ist unterteilt in
 - “A_Dokumente”: Erstellte Dokumente der Phasen Q0 Projektauftrag - Q3 Abnahme
 - “B_Präsentationen”: PowerPoint Präsentationen
 - “C_Protokolle”: Protokolle der Kundenmeetings und Jour Fixe

Umgesetzte Funktionalitäten:

Alle Elemente des Minimal Marketable Features (MMF) wurden umgesetzt:

- Foto aufnehmen
Der User kann in der Webapp ein Foto aufnehmen. Es wird die Kamera des Endgerätes aufgerufen
- Aufgenommenes Foto verwerfen
Der User kann nachdem er das Foto aufgenommen hat dieses verwerfen und erneut ein Foto mit der Kamera seines Endgerätes aufnehmen
- Foto in die Webapp hochladen
Der User kann das aufgenommene Foto in die Webapp hochladen
- Hochgeladenes Foto auf der Landing Page löschen
Der User kann auf der Landing Page das hochgeladene Foto löschen, um erneut ein Foto aufzunehmen und dies dann anschließend in die Webapp hochzuladen
- Verfassen eines Kommentars
Der User kann auf der Landing Page einen Kommentar mit max. 300 Zeichen verfassen, um den Maschinenfehler auf dem Foto zu beschreiben
- Extraktion der Maschinenstammdaten aus dem QR-Code
Die Maschinenstammdaten werden aus dem QR extrahiert und auf der Landing Page angezeigt
- Speicherung der gesamten Downtime-Meldung in einer Datenbank, bestehend aus
 - den Maschinenstammdaten,
 - dem Fotos und
 - dem Kommentars

Weitere Funktionalitäten, die kein MMF waren:

- Fotoupload aus der Galerie
Der User kann ein bereits aufgenommenes Bild, das nicht älter als 12 h sein darf, auswählen und in die Webapp hochladen
- Foto mit Markierungen versehen
 - Markierungen löschen

- bereits hochgeladenes Foto erneut bearbeiten
- bearbeitetes Foto in die Webapp hochladen
- Komprimierung des Fotos auf 5MB
Wird ein Foto in die Webapp hochgeladen, das größer als 5MB ist, so wird dieses auf 5MB herunter skaliert
- Scan Another QR-Code
Der User kann in der Webapp einen weiteren QR-Code scannen, um eine weitere downtime Message zu erfassen und zu dokumentieren. Die Webapp fungiert dann als Kamera
- WaitingPage
Der User kann die Webapp in der WaitingPage offen lassen, um zu einem späteren Zeitpunkt eine weitere downtime Meldung zu erfassen
- Reload der Website
Der Kommentar und das Bild werden im Local Storage gespeichert, sodass bei einem Refresh der Seite die Inhalte nicht verloren gehen

Bekannte Fehler:

- Bis auf ein Improvement wurden alle identifizierten Bugfixes und Verbesserungen aus dem Usability Testessen und den zwei User Tests mit KIEFEL umgesetzt .
Die Möglichkeit im Edit Modus mit zwei Fingern in dem angezeigten Bild zu Zoomen wurde aus zeitlichen Gründen nicht mehr umgesetzt.
- Durch den Systemtest wurden alle Testfälle aus unseren Akzeptanzkriterien der User-Stories vollständig getestet. Dabei ist der eine Testfall nicht erfolgreich durchgelaufen. Der folgende Fehler ist bekannt:

Nach dem Upload eines Fotos größer 5MB in die Webapp wird das Bild komprimiert. Das Bild kann bearbeitet und als eine dtc-Message gespeichert werden, aber ist zu groß, um in den LocalStorage gespeichert zu werden. Refreshed man anschließend die Seite wird das Bild nicht mehr angezeigt, da es nicht im Local Storage gespeichert werden kann.

- [Systemtests Dokumentation](#)

Für nähere Informationen zur Installation, siehe [Installationsanleitung](#)