



WEBANWENDUNG ZUR VISUALISIERUNG VON RETTUNGSRELEVANTEN DATEN ZUR EINSATZUNTERSTÜTZUNG

Software-Entwicklungspraktikum (SEP)

Sommersemester 2023

Pflichtenheft

Auftraggeber

Technische Universität Braunschweig

Peter L. Reichertz Institut für Medizinische Informatik

Prof. Dr. Thomas Deserno

Mühlenpfordtstraße 23

38106 Braunschweig

Betreuer: Viktor Sobotta

Auftragnehmer:

Name	E-Mail-Adresse
Mohamed Wassim Chebili	m.chebili@tu-braunschweig.de
Omar Farouk Khayat	o.khayat@tu-braunschweig.de
Jonas Stepanik	j.stepanik@tu-braunschweig.de
Azhar Rahadian	a.rahadian@tu-braunschweig.de
Kacem Abdennabih	k.abdennabih@tu-braunschweig.de
Torben Oelerking	t.oelerking@tu-braunschweig.de
Qiyue Zhang	qiyue.zhang@tu-braunschweig.de

Braunschweig, 7. Juni 2023

Bearbeiterübersicht

Kapitel	Autoren	Kommentare
1	Torben Oelerking	keine
1.1	Torben Oelerking	keine.
1.2	Torben Oelerking	keine
2	Torben Oelerking	keine
3	Kacem Abdennabih	keine
4	Jonas Stepanik	keine
5	Mohamed Wassim Che- bili	keine
6	Omar Farouk Khayat	keine
6.1	Omar Farouk Khayat	keine
6.2	Omar Farouk Khayat	keine
6.3	Omar Farouk Khayat	keine
6.4	Omar Farouk Khayat	keine
6.5	Mohamed Wassim Che- bili	keine
7	Qiyue Zhang	keine
8	Azhar Rahadian	keine
9	Zusammen	keine

Inhaltsverzeichnis

1 Zielbestimmung	5
1.1 Musskriterien	6
1.2 Sollkriterien	6
1.3 Kannkriterien	7
1.4 Abgrenzungskriterien	7
2 Produkteinsatz	8
2.1 Anwendungsbereiche	8
2.2 Zielgruppen	8
2.3 Betriebsbedingungen	9
3 Produktübersicht	10
4 Produktfunktionen	15
5 Produktdaten	21
6 Nichtfunktionale Anforderungen	23
6.1 Funktionalität	23
6.2 Sicherheit	23
6.3 Benutzbarkeit	24
6.4 Änderbarkeit	25
6.5 Qualitätsanforderungen	25
7 Benutzeroberfläche/Schnittstellen	27
8 Technische Produktumgebung	30
8.1 Software	30
8.2 Hardware	30
8.3 Produktschnittstellen	30
9 Glossar	31

Abbildungsverzeichnis

3.1	Use-Case-Diagramm Aufbau der Webseite	12
3.2	Aktivitätsdiagramm: Unfall visualisieren	13
3.3	Aktivitätsdiagramm: Unfall simulieren	14
5.1	Klassendiagramm Produktdaten	22
7.1	Suche $\langle UI10 \rangle$	27
7.2	Karte $\langle UI20 \rangle$	28
7.3	Suche $\langle UI30 \rangle$	29

1 Zielbestimmung

Um noch mehr Menschenleben in Notfall- und Rettungseinsätzen zu retten, benötigen die Rettungskräfte schnellere und effizientere Entscheidungsprozesse, die ein besseres Verständnis des Unfallgeschehens ermöglichen. Genau hier setzt die zu entwickelnde Webanwendung an - sie wird entwickelt, um Unfälle zu visualisieren und zu simulieren, wobei der Schwerpunkt auf den Daten von beschädigten Fahrzeugen liegt. Bei ausreichend verbleibender Entwicklungszeit ist es erstrebenswert auch die Krafteinwirkung auf Personen zu visualisieren. Auf diese Weise wird versucht die Reaktionsgeschwindigkeit und das Rettungsmanagement von Unfällen unterstützen, zu verbessern und zu beschleunigen.

Um im Vorfeld der Rettungsaktion bereits so viele Daten über den Unfall wie möglich zu erhalten, soll zusätzlich zu klassischer Methode der Datenübermittlung aus Ersthelferaussagen über die Leitstelle auch auf die ISAN (International Standard Accident Number) der in den Unfall verwickelten Fahrzeuge gesetzt werden.

Mit Hilfe der Informationen aus der ISAN stehen Informationen bereit, die das Fahrzeug selbst erfasst hat und sofort im Falle eines Unfalls übertragen kann. Somit stehen verlässliche Informationen über Geschwindigkeiten, Kräfte, Insassenanzahl, Fahrzeugmodell und die Fahrzeugklasse (z.B. Geländewagen, Kombi, Limousine) zum Zeitpunkt des Unfalls zur Verfügung, die aufgeregte und oder geschockte Ersthelfer im entsprechenden Moment nur schwer erfassen oder an der Unfallstelle falsch einschätzen können.

Die Webanwendung soll die Machbarkeit eines solchen Systems demonstrieren und diese übermittelten Daten erfassen und aufbereiten. Der gewünschte Unfall wird über eine Unfall-ID, die in ein Suchfeld eingegeben wird, aufgerufen. Der Unfallschaden des Fahrzeuges wird simuliert und ein entsprechendes 3D-Modell auf einer Karte an den Koordinaten des Unfallortes angezeigt. Eine Verlinkung zur entsprechenden Rettungskarte des Fahrzeuges liefert rettungsrelevante Informationen für die Einsatzkräfte. Des Weiteren könnten in einer fortgeschrittenen Ausbaustufe Informationen über die Krafteinwirkungen auf die Insassen abgerufen werden, aus denen dann vom Anwender auf die Verletzungen geschlossen werden kann.

Mit Hilfe dieser Zusatzinformationen kann sich bereits während der Fahrt zum Unfallort die Einsatzleitung ein Bild des Unfalls machen und eine erste Planung des Einsatz durchführen. Dank der Simulation und der zum Fahrzeug passenden Rettungskarte soll es möglich sein, be-

reits im Vorfeld, ohne das reale Fahrzeug gesehen zu haben, eine Insassenrettung inklusive der nötigen Schritte und ggf. Schnitte zu Planen.

Die rettungsrelevanten Daten zur Planung der Rettungsaktion lassen sich mit der Webanwendung schnell, intuitiv und visuell einfach erfassen. Mögliche Rückfragen der Einsatzleitung an die Leitstelle während der Fahrt beschleunigen die Rettung enorm. Des Weiteren wird die Zeit, die benötigt wird um sich am Unfallort einen Überblick zu verschaffen, reduziert. Die Aufgabenverteilung, Zuweisung und Unterrichtung der Einsatzkräfte über das Lagebild kann ebenfalls bereits während der Fahrt erfolgen, sodass die Rettung am Unfallort direkt starten kann!

1.1 Musskriterien

Folgende Funktionalitäten und Leistungen muss das Softwareprodukt erfüllen:

- ⟨RM1⟩ Die Anwendung muss ISANs entgegennehmen können und die Daten auswerten können
- ⟨RM2⟩ Die ISANs werden gespeichert. Zeitlich und geografisch nahe beieinanderliegende Nachrichten werden zu einem Unfall mit eindeutiger ID zusammengefasst
- ⟨RM3⟩ Der Schaden des Fahrzeuges wird mittels BeamNG und den Informationen aus der ISAN auf einem Server simuliert
- ⟨RM4⟩ Die Simulation des Unfalls liefert 3D-Modelle der beschädigten Fahrzeuge zurück.
- ⟨RM5⟩ Es gibt eine Website mit eingebetteter Karte, die die Modelle entgegennimmt und an den Koordinaten des Unfalls anzeigt.
- ⟨RM6⟩ Es gibt eine Startseite mit einem Suchfeld, in welches die ID des Unfalls eingegeben werden kann und man dann die Kartenansicht mit den Fahrzeugen des Unfalls und weiteren Informationen erhält.
- ⟨RM7⟩ Ein Unfall wird vom Server nach einer fest definierten Zeit gelöscht. Damit werden alle dazugehörigen Daten und Modelle von der Karte entfernt und auf dem Server gelöscht.
- ⟨RM8⟩ Es wird zum ausgewählten Fahrzeug ein Link zu der entsprechenden Rettungskarte, die sich in einem neuen Fenster öffnet, bereitgestellt.

1.2 Sollkriterien

Folgende Funktionalitäten und Leistungen sind erstrebenswert:

- ⟨RS1⟩ Das Erscheinungsbild der Website soll ansprechend, übersichtlich, anwenderfreundlich und intuitiv gestaltet sein.
- ⟨RS2⟩ Wenn der Mauszeiger sich über einem Fahrzeug befindet, soll ein Fenster eingeblendet

werden mit Informationen zu Krafteinwirkung und Insassenanzahl

1.3 Kannkriterien

Folgende Funktionalitäten und Leistungen können implementiert werden:

- ⟨RC1⟩ Bei der Auswahl eines Fahrzeuges werden die Krafteinwirkungen als Heatmap/Vertex auf die Insassen angezeigt
- ⟨RC2⟩ Die Anzeige der Krafteinwirkungen auf die Insassen erfolgt entweder abhängig vom ausgewählten Fahrzeug neben der Karte oder wird als extra Fenster eingeblendet.
- ⟨RC3⟩ Die Darstellungen der Insassen mit Krafteinwirkung sollen sich drehen lassen, sodass sich auch die Krafteinwirkung auf der Rückseite der Figuren erkennen lässt.
- ⟨RC4⟩ Die Verlinkung zur Rettungskarte kann entweder für jedes Fahrzeug des Unfalls fest eingeblendet werden oder es wird nur die Verlinkung passend zum aktuell ausgewählten Fahrzeug angezeigt oder die Verlinkung ist in dem Fenster mit Kraft und Insassenanzahl enthalten, welches sich öffnet, wenn der Mauszeiger über dem Fahrzeug schwebt.

1.4 Abgrenzungskriterien

Folgende Funktionalitäten und Leistungen müssen nicht explizit umgesetzt werden:

- ⟨RW1⟩ Ein Handbuch oder Bedienungsanleitung ist nicht im Projektumfang enthalten.
- ⟨RW2⟩ Das Projekt wird nicht nach Anforderungen für Software für sicherheitskritische Anwendungen entwickelt. Das Projekt soll zeigen, dass sich die Grundidee umsetzen lässt.
- ⟨RW3⟩ An dem Projekt der Rettungskarte wird nicht gearbeitet. Dieses wird von Seiten des PLRIs zur Verfügung gestellt. Es wird nur eine Verlinkung in der neu zu entwickelnden Anwendung geben.
- ⟨RW4⟩ Das System wird nicht darauf ausgelegt sein, dass eine große Anzahl von Nutzern gleichzeitig versucht darauf zuzugreifen.
- ⟨RW5⟩ Ein Unfall wird von dem System selbst gelöscht. Es wird keine Möglichkeit geben, dass von außen ein Unfall gelöscht werden kann.

2 Produkteinsatz

2.1 Anwendungsbereiche

Die zu entwickelnde Softwareanwendung ist für die Visualisierung rettungsrelevanter Daten bei Fahrzeugunfällen gedacht, sodass diese bei Fahrzeugunfällen zur Rettungsunterstützung eingesetzt werden kann.

Durch die Beschränkung der Visualisierung auf die Darstellung der Unfallfahrzeuge auf einer Karte und in einer weiteren Ausbaustufe mit der Visualisierung der Krafteinwirkung auf die Insassen, eignet sich die Anwendung nur für die Unfallrettung mit Fahrzeugbeteiligung.

Durch zusätzliche Informationen die pro Fahrzeug zur Verfügung gestellt werden, z.B. die maximal wirkende Kraft im Unfallmoment oder die Rettungskarte kann die Anwendung bereits auf dem Weg zum Unfallort genutzt werden kann und eine mögliche Rettungsplanung vorab erledigt werden.

Die Anwendung ist nicht für die Rekonstruktion des Unfallhergangs oder für die Klärung von Schuldfragen durch Polizei oder Versicherungen geeignet, da die Fahrzeuge nur an der endgültigen Unfallposition angezeigt werden und somit keine Vorgeschichte zum Unfallhergang existiert. Des Weiteren werden die Daten nach fest definierter Zeit automatisiert gelöscht.

Da es sich aber bei dem Projekt um ein Teilprojekt des ISAN-Projektes handelt und der gesamte Quellcode mit Projektende dem Auftraggeber zur Verfügung gestellt wird, ist es möglich bei fortschreitender Entwicklung des ISAN-Projektes unser Projekt mit einzubinden und weiterzuentwickeln und den Funktionsumfang an neu entstandene Bedürfnisse anzupassen und so neue Anwendungsbereiche zu erschließen.

2.2 Zielgruppen

Die Webanwendung zur Simulation und Visualisierung von Autounfällen und der Bereitstellung von Informationen richtet sich an Kreise und Kommunen, die ihre Rettungskräfte und deren Einsatzleitung mit einer modernen, effizienten Software ausstatten wollen, um in Unfallsituationen Leben zu retten, Einsatzzeiten zu reduzieren, den Bürger noch besser schützen zu können und lange Ausfallzeiten von Verkersadern zu reduzieren.

Unsere Webanwendung ist für erfahrene Einsatzleitungen gedacht, die die Anwendung nutzen, um vor Ort schnell auf Unfälle reagieren zu können. Durch die Nutzung unserer Webanwendung erhalten sie bereits vor der Ankunft am Unfallort einen Eindruck von der Lage, was zu einer schnelleren Reaktion führt. So kann bereits während der Fahrt vom Fahrzeug aus mit Hilfe der visualisierten Daten der Einsatz geplant werden.

2.3 Betriebsbedingungen

Folgende Bedürfnisse und Anforderungen werden an das Produkt gestellt:

- Die Softwareanwendung soll im Browser bzw. im Einsatz auf dem tragbaren NidaPad laufen, um den physikalischen Bedingungen im mobilen Einsatz gewachsen zu sein. Das NidaPad wird von den Rettungskräften im Einsatz, aber auch bereits auf dem Weg zum Unfallort im Fahrzeug verwendet. Das NidaPad eignet sich auf Grund seiner robusten Bauweise für die vielfältigen und ggf. harschen Einsatzumgebungen am Unfallort.
- Da die Softwareanwendung später einmal rund um die Uhr in der Unfallrettung eingesetzt werden soll und ISANs von Unfallfahrzeugen entgegen nehmen soll, muss die Anwendung im realen Einsatz zuverlässig im Dauerbetrieb laufen. Dies bedeutet, dass sowohl die Webanwendung im Browser im Zusammenspiel mit dem Nidapad zuverlässig rund um die Uhr funktionieren muss.

Aber auch der Server auf dem die ISANs eingehen, verarbeitet werden und die Simulation der Unfallfahrzeuge stattfindet, muss, sowie der Webserver, der die Webanwendung bereitstellt, zuverlässig 24 Stunden, sieben Tage die Woche funktionieren und erreichbar sein.

3 Produktübersicht

In diesem Abschnitt werden die Funktionalitäten der zu entwickelnden Webseite sowohl grafisch mithilfe von Use-Case-Diagrammen als auch durch einen kurzen Text beschrieben. Im Text wird deutlich, welches Kriterium jeder Use Case erfüllt. Es sind alle erforderlichen Use-Case-Diagramme vorhanden, um die Funktionalitäten vollständig zu beschreiben. Der Abschnitt „Unfall simulieren“ wird detaillierter in einem Aktivitätsdiagramm dargestellt, um die einzelnen Schritte, die durchlaufen werden, besser zu beschreiben.

Abbildung 3.1

Im Use-Case-Diagramm „Aufbau der Webseite“ wird die allgemeine Struktur der Webseite dargestellt, sowohl von der Nutzerseite als auch von der Serverseite aus betrachtet.

Nachdem der Nutzer die Startseite betritt, stehen ihm zwei verschiedene Punkte zur Auswahl. Wenn er eine ID eingibt, wird der zugehörige Unfall auf der Karte angezeigt.

Ein weiterer Menüpunkt ist die Karte, auf der der Nutzer den Unfall auf der Karte, die Krafteinwirkungen auf die Insassen und die Rettungskarte betrachten kann. Darüber hinaus hat er die Möglichkeit, den ausgewählten Unfall zu vergrößern, die Karte zu verkleinern, um einen allgemeinen Überblick zu erhalten, oder die Karte zu verschieben.

Sobald ein Unfall passiert ist, überträgt den Server die zugehörigen ISANs (Internationale Standardunfallnummer) zusammen mit den relevanten Unfalldaten von den am Unfall beteiligten Fahrzeugen an das System. Nachdem die ISAN erfolgreich ausgewertet und als korrekt befunden wurde, werden alle relevanten Daten extrahiert und auf dem Server gespeichert. Nachdem die ISAN erfolgreich ausgewertet und als korrekt Form vorliegt, werden alle relevanten Daten ausgelesen und auf dem Server gespeichert. Nach einer bestimmter Zeit wird der Unfall vom System gelöscht. Andernfalls wird der Unfall vom Simulationsserver simuliert. Nach Abschluss der Simulation erfolgt die Darstellung in Form einer Visualisierung. Der Web-Server erhält die erfolgreiche response von der Simulation-Server und aus der response werden visualisierungsrelevante Daten ausgelesen und gespeichert und dann wird das 3D-Mesh auf der Karte angezeigt. Wenn der Nutzer auf den „Rettungskarte“-Button klickt, wird die entsprechende ISAN ermittelt. Anschließend wird nach dem Link zur Rettungskarte gesucht und diese in einem neuen Fenster geöffnet. Dazu kann der Nutzer auch die Verletzungen in dem zugehörigen Fahrzeug visualisieren, das heißt bei der Auswahl eines Fahrzeuges werden die Krafteinwirkungen als Heatmap/Vertex

auf die Insassen angezeigt. Zu guter Letzt bietet das System dem Nutzer die Visualisierung des Unfalls auf einer Webseite an.

Abbildung 3.2

Das Aktivitätsdiagramm „Unfall visualisieren“ veranschaulicht den Ablauf der Visualisierung eines Unfalls, der vom Server durchgeführt wird. Zunächst wird überprüft, ob die Unfallsimulation bereits durchgeführt wurde.

Falls die Simulation noch nicht erfolgt ist, wird die Simulation des Unfalls gestartet.

Danach werden aus der Response die Daten, die für die Visualisierung relevant sind, ausgelesen und gespeichert. Schließlich wird das 3D-Mesh auf der Karte angezeigt, um den Unfall visuell darzustellen.

Abbildung 3.3

Das Aktivitätsdiagramm „Unfall simulieren“ veranschaulicht den Ablauf der Simulation eines Unfalls, die vom Server durchgeführt wird. Zunächst wird überprüft, ob eine ISAN (Internationale Standardunfallnummer) vorliegt und ausgewertet wurde.

Falls die ISAN nicht vorhanden ist, muss eine Auswertung der ISAN erfolgen. Dazu wird ein GET-REQUEST an den Simulations-Server gesendet, um relevante Daten für die Simulation zu übermitteln.

Danach wird die Simulation mit den übermittelten Daten gestartet, um die relevanten Szenarien nachzustellen. Sobald die Simulation abgeschlossen ist, werden die Ergebnisse der Simulation ausgelesen.

Schließlich werden die Ergebnisse der Simulation als Response zurückgegeben.

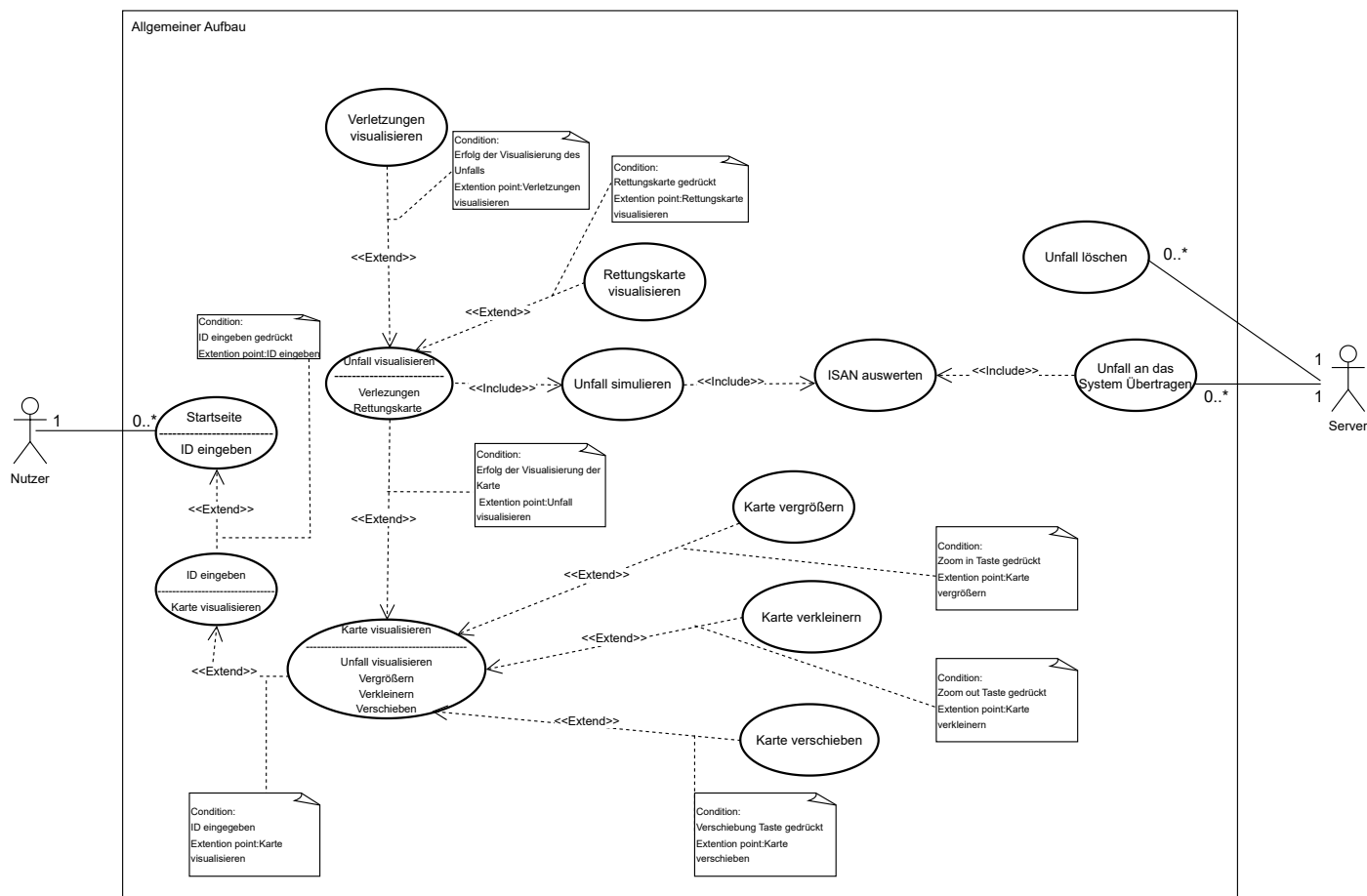


Abbildung 3.1: Use-Case-Diagramm Aufbau der Webseite

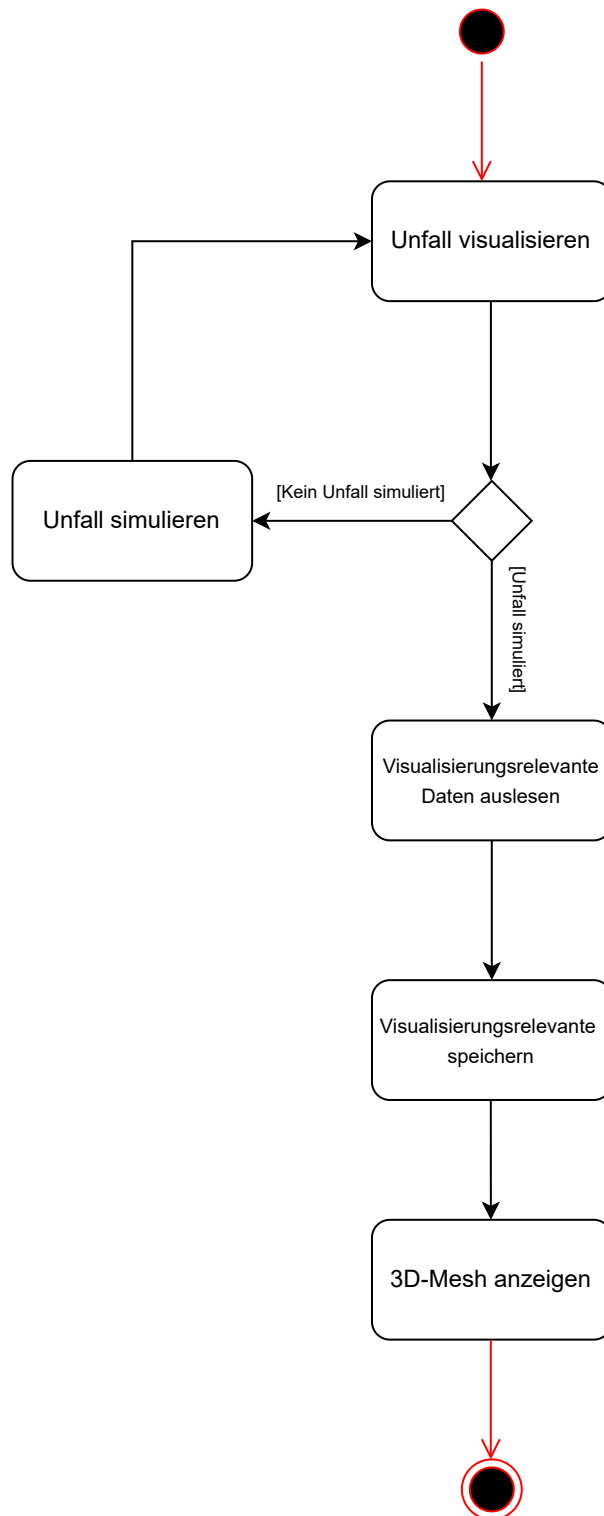


Abbildung 3.2: Aktivitätsdiagramm: Unfall visualisieren

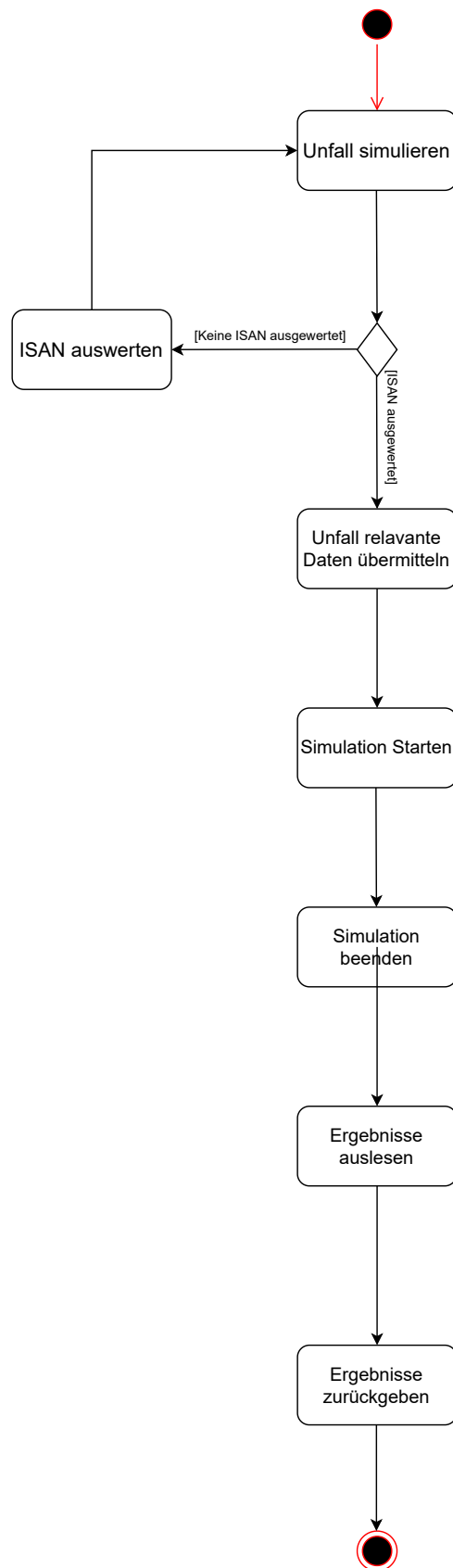


Abbildung 3.3: Aktivitätsdiagramm: Unfall simulieren

4 Produktfunktionen

Unsere Software zur Simulation von Autounfällen und Bereitstellung von Informationen für Rettungskräfte bietet eine Vielzahl leistungsstarker Funktionen, um den Einsatzkräften bei der effizienten Planung und Durchführung von Rettungsmaßnahmen zu unterstützen.

Auswertung einer ISAN $\langle F1 \rangle$

Anwendungsfall: Übertragung eines Unfalls an das System.

Anforderung: $\langle RM1 \rangle$

Ziel: Aus der ISAN werden alle relevanten Daten ausgelesen und auf dem Server gespeichert.

Vorbedingungen: Die ISAN liegt im korrekten Format vor.

Nachbedingungen Erfolg: Alle Daten aus der ISAN sind separat auf dem Server zur Weiterverarbeitung gespeichert. Die REST-API des Web-Servers signalisiert Erfolg als response.

Nachbedingungen Fehlschlag: Es werden keine Daten aus der ISAN gelesen. Die REST-API signalisiert Fehlschlag als response.

Akteure: Web-Server

Auslösendes Ereignis: Auf dem Web-Server geht eine POST-REQUEST ein.

Zusammengehörende Unfälle werden gruppiert $\langle F2 \rangle$

Anwendungsfall: Es gehen mehrere Unfälle als ISAN's in das System ein.

Anforderung: $\langle RM2 \rangle$

Ziel: Mehrere ISAN's, welche zu einem gemeinsamen größeren Unfall gehören werden zusammen gruppiert. ISAN's, welche nicht zusammengehören, werden nicht einander zugeordnet.

Vorbedingungen: Es liegen mehrere ausgewertete ISAN's (Erfolg von $\langle F1 \rangle$) im System vor.

Nachbedingungen Erfolg: Mehrere ISAN's sind gemeinsam unter einer Unfall-ID einsehbar.

Nachbedingungen Fehlschlag: Es werden keine ISAN's gruppiert. Jede ISAN erhält eine eigene Unfall-ID.

Akteure: Web-Server

Auslösendes Ereignis: Es wird dem System eine neue ISAN übermittelt.

Beschreibung:

1. Es wird eine ISAN übermittelt und ausgewertet.
2. Rufe die zeitlichen und geografischen Informationen der ISAN auf.
3. Fasse mit ISAN's zusammen, welche zeitlich und geografisch nah beieinander liegen.
4. Der Server gibt eine erfolgreiche Response zusammen mit der Unfall-ID zurück.

Simulation eines Unfalls $\langle F3 \rangle$

Anwendungsfall: Eine ISAN wurde ausgewertet.

Anforderung: $\langle RM3 \rangle$ $\langle RM4 \rangle$

Ziel: Ein Unfall wird korrekt simuliert und an den Web-Server übertragen. Insbesondere wird dabei ein 3D-Modell des beschädigten Fahrzeugs übermittelt.

Vorbedingungen: Erfolg von $\langle F1 \rangle$

Nachbedingungen Erfolg: Alle Daten aus der erfolgreichen Simulation sind gespeichert und werden an den Web-Server zurückgegeben. Die REST-API des Simulations-Servers signalisiert Erfolg als response.

Nachbedingungen Fehlschlag: Der Simulations-Server führt keine Erfolgreiche Simulation aus und reicht Informationen aus der Simulation nicht zurück. Die REST-API des Simulations-Servers signalisiert Fehlschlag als response.

Akteure: Simulations-Server

Auslösendes Ereignis: Der Web-Server übermittelt Daten eines Unfalls an den Simulations-Server und erwartet Ergebnisse aus der Simulation zurück.

Beschreibung:

1. Es geht eine GET-REQUEST auf dem Simulations-Server ein, welche relevante Daten für die Simulation übermittelt.
2. Für die Daten wird eine relevante Simulation gestartet.
3. Sobald die Simulation beendet ist, werden Ergebnisse der Simulation ausgelesen.
4. Ergebnisse der Simulation werden als response zurückgegeben.

Visualisierung eines Unfalls auf der Karte $\langle F4 \rangle$

Anwendungsfall: Übertragung eines Unfalls an das System und daraufhin erfolgreiche Simulation des Unfalls.

Anforderung: $\langle RM5 \rangle$

Ziel: Das 3D-Mesh des simulierten Unfalls wird korrekt auf der Karte angezeigt.

Vorbedingungen: Erfolg von $\langle F3 \rangle$

Nachbedingungen Erfolg: Auf einer Karte wird ein 3D-Modell des kaputten Fahrzeugs an der richtigen Stelle angezeigt.

Nachbedingungen Fehlschlag: Auf der Karte wird kein 3D-Modell angezeigt.

Akteure: Web-Server

Auslösendes Ereignis: Der Simulations-Server liefert ein 3D-Modell aus einer Simulation zurück.

Beschreibung:

1. Der Web-Server erhält die erfolgreiche response von $\langle F3 \rangle$
2. Aus der response werden visualisierungsrelevante Daten ausgelesen und gespeichert.
3. Das 3D-Mesh wird auf der Karte angezeigt.

Suchen eines Unfalls $\langle F5 \rangle$

Anwendungsfall: Dauerbetrieb

Anforderung: $\langle RM6 \rangle$

Ziel: Für die jeweilige Unfall-ID werden auf einer Karte die relevanten Autos angezeigt.

Vorbedingungen: -

Nachbedingungen Erfolg: Die eingegebene Unfall-ID existiert und es werden gegebenenfalls mehrere zugehörige Unfälle gefunden und auf einer Karte angezeigt.

Nachbedingungen Fehlschlag: Die eingegebene Unfall-ID existiert nicht. Man kann eine neue Unfall-ID eingeben.

Akteure: Web-Server

Auslösendes Ereignis: Es wird eine Unfall-ID in ein Suchfeld eingegeben.

Beschreibung:

- Es wird eine Unfall-ID in das Suchfeld eingegeben.
- Die Unfall-ID wird in einer Datenbank, welche alle Unfälle speichert, mittels einer linearen Suche gesucht.

- Wird der Unfall gefunden, so werden alle zugehörigen ISAN's zusammen mit ihren Simulationsergebnissen ermittelt.
- Alle zugehörigen Autos werden auf einer Karte angezeigt.

Löschen eines Unfalls $\langle F6 \rangle$

Anwendungsfall: Ein Unfall ist abgelaufen.

Anforderung: $\langle RM7 \rangle$

Ziel: Nach einer fest definierten Zeit wird der Unfall und damit alle zugehörigen ISAN's, Modelle und Daten vom Server gelöscht.

Vorbedingungen: Es existiert mindestens ein Unfall auf dem Server.

Nachbedingungen Erfolg: Der Unfall und damit alle zugehörigen ISAN's, Modelle und Daten sind vom Server gelöscht.

Nachbedingungen Fehlschlag: Der Unfall und damit alle zugehörigen ISAN's, Modelle und Daten sind nicht vom Server gelöscht.

Akteure: Web-Server

Auslösendes Ereignis: Der Unfall existiert bereits einen fest definierten Zeitraum auf dem Server.

Beschreibung:

1. Die Datenbank aller Unfälle wird nach Unfällen durchsucht, welche gelöscht werden sollen, weil sie schon einen fest definierten Zeitraum existieren.
2. Es wird ein entsprechender Unfall gefunden.
3. Der Unfall und damit alle zugehörigen ISAN's, Modelle und Daten werden vom Server gelöscht.

Aufrufen der Rettungskarte $\langle F7 \rangle$

Anwendungsfall: Der Nutzer möchte weitere unfallrelevante Daten zu einem bestimmten Auto abrufen.

Anforderung: $\langle RM8 \rangle$

Ziel: Die Rettungskarte wird in einem neuen Fenster angezeigt.

Vorbedingungen: Es existiert ein Fahrzeug und die zugehörige Rettungskarte ist bekannt.

Nachbedingungen Erfolg: Die Rettungskarte wird angezeigt.

Nachbedingungen Fehlschlag: Die Rettungskarte wird nicht angezeigt.

Akteure: Web-Server

Auslösendes Ereignis: Der Nutzer klickt auf einen Button bei dem entsprechenden Fahrzeug.

Beschreibung:

1. Der Nutzer klickt den Button.
2. Zu dem Fahrzeug wird die entsprechende ISAN ermittelt und damit der Link zur Rettungskarte gesucht.
3. Der Link wird in einem neuen Fenster geöffnet.

Anzeige der Fahrzeuginformationen $\langle F8 \rangle$

Anwendungsfall: Der Nutzer möchte sich zu einem Unfallfahrzeug weitere Informationen anzeigen lassen, wie Krafteinwirkung oder Insassenzahl.

Anforderung: $\langle RS2 \rangle$

Ziel: Es wird ein Fenster eingeblendet, welche die relevanten Informationen anzeigt.

Vorbedingungen: Es existiert ein Fahrzeug und der Nutzer befindet sich mit dem Mauszeiger über jenem Fahrzeug.

Nachbedingungen Erfolg: Ein Fenster wird angezeigt, welches die relevanten Informationen präsentiert.

Nachbedingungen Fehlschlag: Es wird kein Fenster angezeigt.

Akteure: Web-Server

Auslösendes Ereignis: Der Nutzer fährt mit dem Mauszeiger über ein Fahrzeug.

Beschreibung:

1. Der Nutzer fährt mit dem Mauszeiger über ein Fahrzeug.
2. Das darunterliegende Fahrzeug wird ermittelt und damit die zugehörigen Informationen, welche auf dem Server gespeichert sind.
3. Es wird ein Fenster neben dem Fahrzeug mit den relevanten Informationen angezeigt.

Anzeige der Krafteinwirkung auf Insassen $\langle F9 \rangle$

Anwendungsfall: Der Nutzer möchte sich zu einem Unfallfahrzeug die Krafteinwirkung auf die Insassen im Fahrzeug anzeigen lassen.

Anforderung: $\langle RC1 \rangle$

Ziel: In einem separaten Fenster werden die einzelnen Insassen mit Heatmap zur Krafteinwirkung angezeigt.

Vorbedingungen: Es existiert ein Unfall und dieser ist auf der Karte geöffnet.

Nachbedingungen Erfolg: In einem Fenster werden die zuvor aufgeführten Informationen präsentiert.

Nachbedingungen Fehlschlag: Es wird kein Fenster angezeigt.

Akteure: Web-Server

Auslösendes Ereignis: Der Nutzer wählt ein Fahrzeug aus

Beschreibung:

1. Der Nutzer wählt ein Fahrzeug aus.
2. Die Informationen zu den Krafteinwirkungen auf die Insassen dieses Fahrzeugs werden abgerufen. Diese Informationen sind in einem vorherigen Schritt aus der Simulation gekommen.
3. In einem separaten Fenster werden bei Erfolg die Insassen mit entsprechender Heatmap angezeigt.
4. Bei Fehlschlag soll gar kein Fenster angezeigt werden, dementsprechend auch keine Informationen über die Insassen.

5 Produktdaten

Unsere Anwendung benötigt Daten, um ordnungsgemäß zu funktionieren. Diese Daten lassen sich in folgende Kategorien einteilen:

Unfall $\langle D1 \rangle$

Daten des Unfalls :

- Uhrzeit
- Datum
- Id
- Autos $\langle D2 \rangle$

Auto $\langle D2 \rangle$

Daten des Autos :

- Geografischer Länge
- Geografischer Breite
- 3D Mesh
- Kfz-Kennzeichen
- HSN/TSN
- Kraft
- Geschwindigkeit
- Anzahl Passagiere

Die Relation zwischen diese Daten lassen sich mit folgendem Klassendiagramm darstellen:

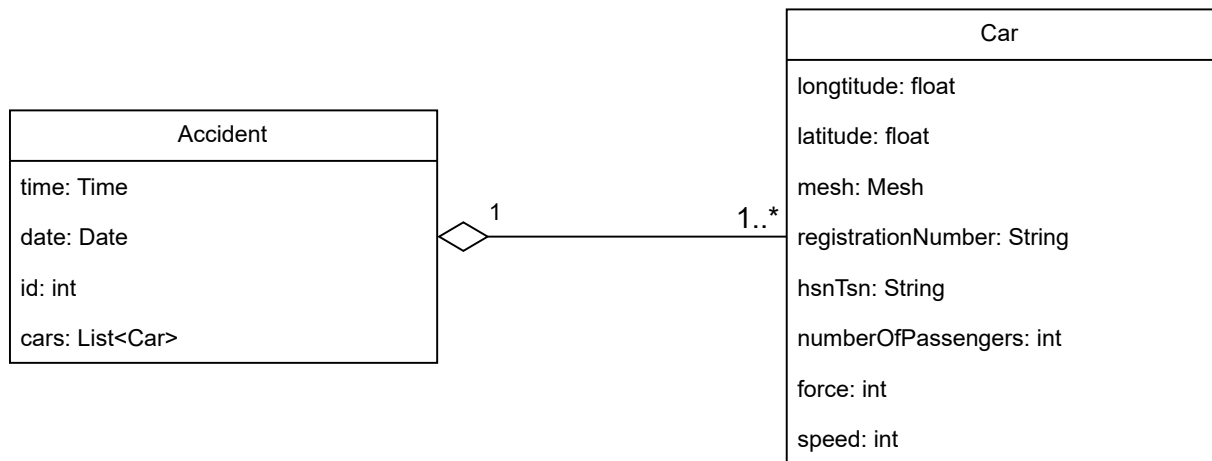


Abbildung 5.1: Klassendiagramm Produktdaten

6 Nichtfunktionale Anforderungen

In diesem Abschnitt werden die nichtfunktionalen Anforderungen und Qualitätsmerkmale der Software beschrieben. Dabei werden Entscheidungen hinsichtlich der wichtigsten Merkmale getroffen und in einer Tabelle detailliert beschrieben.

6.1 Funktionalität

Produktqualität	sehr gut	gut	normal	nicht relevant
Angemessenheit	x			
Richtigkeit	x			
Interoperabilität	x			
Ordnungsmäßigkeit				x

Angemessenheit : Es ist erforderlich, dass die Anwendung alle vorgegebenen Funktionen erfüllt.

Richtigkeit : Richtigkeit ist wichtig zu betrachten, da sie direkt Einfluss auf die Notfallreaktion hat und die Fähigkeit der Einsatzkräfte beeinflusst, genaue Informationen korrekt abzurufen.

Interoperabilität : Es ist unerlässlich, dass unsere Anwendung in der Lage ist, die ISAN-Nummer eines Fahrzeugs abzurufen. Darüber hinaus ist es wichtig, dass der Client die Informationen und simulierten Mechanismen vom Server empfangen kann. Aus diesem Grund ist die Interoperabilität ein entscheidendes Element unserer Anwendung.

Ordnungsmäßigkeit : Unser Produkt wird nicht als endgültiges Produkt entwickelt. Datenschutz und andere Faktoren müssen zu einem späteren Zeitpunkt berücksichtigt werden

6.2 Sicherheit

Produktqualität	sehr gut	gut	normal	nicht relevant
Zuverlässigkeit	x			
Reife		x		
Fehlertoleranz		x		
Wiederherstellbarkeit		x		

Zuverlässigkeit : Die Webanwendung wird von Einsatzkräften genutzt, um den Ort eines Unfalls zu lokalisieren und relevante Daten wie Verletzungen und Rettungskarten einzusehen. Es ist wichtig, dass unsere Software diese Funktionen zuverlässig ausführen kann

Reife : Die Reife der Software bedeutet, dass weniger Fehler auftreten können. Dies ist von entscheidender Bedeutung, da ein Fehler zu einem fehlgeschlagenen Einsatz führen kann.

Fehlertoleranz : Es ist wichtig, Fehler zu vermeiden. Allerdings sollte unsere Software auch dann korrekt arbeiten, wenn zum Beispiel die Verarbeitung eines Unfalls fehlschlägt. In diesem Fall sollte die Karte weiterhin alle anderen Unfälle fehlerfrei anzeigen.

Wiederherstellbarkeit : Wenn die Webanwendung nicht in der Lage ist, neue Daten vom Server zu laden, muss sie die vorhandenen Daten beibehalten können. Es ist wichtig, dass die Webanwendung auch in diesem Fall ordnungsgemäß funktioniert.

6.3 Benutzbarkeit

Produktqualität	sehr gut	gut	normal	nicht relevant
Verständlichkeit	x			
Erlernbarkeit			x	
Bedienbarkeit		x		
Effizienz	x			
Zeitverhalten		x		
Verbrauchsverhalten				x

Verständlichkeit : Die Informationen und Daten müssen für die Einsatzkräfte klar und verständlich sein, um eine effektive Bewältigung der Situation zu gewährleisten

Erlernbarkeit : Die Webanwendung ist nicht für den normalen Nutzer gedacht, sondern soll von geschulten Einsatzkräften bedient werden.

Bedienbarkeit : Die Webanwendung muss benutzerfreundlich gestaltet sein, um Verzögerungen bei der Bedienung zu vermeiden.

Effizienz : Da die Webanwendung gelegentlich viele 3D-Objekte auf einer Karte anzeigen muss, ist es wichtig, dass sie effizient programmiert wird, um Verzögerungen oder langsame Software zu vermeiden.

Zeitverhalten : Die Webanwendung muss in Echtzeit Unfalldaten verarbeiten und diese schnell auf der Benutzeroberfläche anzeigen, um eine schnelle Reaktion zu ermöglichen.

Verbrauchsverhalten : Die Webanwendung wird auf einem geeigneten PC laufen und die Ressourcen des PCs können vollständig genutzt werden. Daher ist das Verbrauchsverhalten als weniger relevant anzusehen.

6.4 Änderbarkeit

Produktqualität	sehr gut	gut	normal	nicht relevant
Analysierbarkeit		x		
Modifizierbarkeit		x		
Stabilität	x			
Prüfbarkeit		x		
Übertragbarkeit				x
Anpassbarkeit		x		
Installierbarkeit				x
Konformität			x	
Austauschbarkeit		x		

Analysierbarkeit und Prüfbarkeit : Eine gewisse Prüfbarkeit und Analysefähigkeit sind wichtig, um sicherzustellen, dass die Anwendung korrekt funktioniert, die notwendigen Funktionen erfüllt und Fehler schnell identifiziert und behoben werden können.

Modifizierbarkeit, Austauschbarkeit und Anpassbarkeit : Die Software sollte erweiterbar sein, um zukünftige Funktionen hinzuzufügen oder vorhandene zu ersetzen. Zum Beispiel könnten spezielle Einsatzdienste oder neue Daten, die von zukünftigen Autos bereitgestellt werden, geeignete Funktionen erfordern. Daher muss die Software austauschbar, modifizierbar und anpassbar sein.

Stabilität : Die Webanwendung müssen stabil arbeiten, um die Daten korrekt und zeitnah liefern zu können.

Übertragbarkeit und Installierbarkeit : Die Software muss auf den geeigneten Geräten, auf denen sie getestet wurde, einwandfrei funktionieren. Es besteht keine Notwendigkeit, dass sie auf vielen verschiedenen Plattformen arbeitet.

Konformität : Die Software muss auf verschiedenen unterstützten Geräten und Browsern konsistente Ergebnisse liefern, um eine hohe Konformität sicherzustellen.

6.5 Qualitätsanforderungen

Die als am wichtigsten eingestuften Qualitätsmerkmale werden nun operationalisiert, indem detaillierte Produktanforderungen festgelegt werden und die einzuhaltenden Richtlinien (z.B. Standards und Normen) angegeben werden

- $\langle Q10 \rangle$ Die Funktion $\langle F3 \rangle$ sollte benutzerfreundlich gestaltet und einfach zu bedienen sein.
- $\langle Q20 \rangle$ Die Sprache der Webanwendung soll deutsch sein
- $\langle Q30 \rangle$ Die Informationen auf der Webanwendung sollen verständlich sein .
- $\langle Q40 \rangle$ Das Produkt soll auf übliche Browser laufen.
- $\langle Q50 \rangle$ Das Produkt soll in der Lage sein, 3D-Objekte auf einer Karte anzuzeigen, ohne dass die Systemleistung beeinträchtigt wird.
- $\langle Q60 \rangle$ Das Produkt soll stabil laufen.
- $\langle Q70 \rangle$ Die Korrektheit der Datenübertragung vom Server zum Client soll überprüft werden.
- $\langle Q80 \rangle$ Der Code soll ausreichend kommentiert und klar modularisiert sein.
- $\langle Q90 \rangle$ Die Übertragung des Unfalls auf die Karte sollte innerhalb von 3 Sekunden erfolgen.

7 Benutzeroberfläche/Schnittstellen

In diesem Kapitel wird die Benutzeroberfläche beschrieben. Es wird die betroffenen Autounfällen und relevante Daten über die Fahrgäste und Verkehr in den Schnittstellen gezeigt.

Benutzeroberfläche:

Bei dem Projekt handelt es sich um eine Webanwendung, die auf dem Nidapad laufen und die Visualisierung von unterschiedlichen Daten der Autounfällen, z.B. die Insassen, auf der Website darstellen soll.

Diese Webanwendung wird sowohl vor Ort als auch in Rettungsfahrzeugen und Einsatzfahrzeugen genutzt, daher ist es von Bedeutung, dass die Struktur der Benutzerschnittstelle möglichst klar und einfach gestaltet ist, damit die Anwender während des Einsatzes nicht verwirrt werden. Die Benutzerschnittstelle sollte eine klare Hierarchie und Beschriftung von Elementen aufweisen und auf überflüssige visuelle Effekte verzichten.

Suche $\langle UI10 \rangle$

Am Anfang wird der Benutzer auf der Suchseite landen und wird aufgefordert, eine Unfall ID in den Suchfeld einzugeben. Nach dem anschließenden Klick auf den Bestätigungsknopf wird der Nutzer zu der Kartenansicht weitergeleitet und die Visualisierung des entsprechenden Unfalls auf Karte angezeigt.



The image shows a web interface titled "Suche" (Search). Below the title is a search input field with a magnifying glass icon on the left and the placeholder text "Unfall_ID eingeben". The entire interface is enclosed in a rectangular border.

Abbildung 7.1: Suche $\langle UI10 \rangle$

Kartenansicht $\langle UI20 \rangle$

In der Kartenansicht wird eine Übersichtskarte präsentiert, die die exakte Unfallposition sowie alle involvierten Fahrzeuge abbildet. Sobald der Benutzer den Mauszeiger über ein Fahrzeug bewegt, erscheint ein Informationsfeld, das detaillierte Angaben über das Fahrzeug bereitstellt, hierzu gehören die Anzahl der Insassen, sowie die maximale Kraft, die während des Unfalls auf das Fahrzeug wirkte. Durch Anklicken eines Fahrzeugs werden auf der linken Seite der Webseite die Kraftwirkungen für die jeweiligen Insassen dargestellt. Am unteren Ende der Seite ist eine Liste von verlinkten Rettungskarten für die in den Unfall verwickelten Fahrzeuge zu finden. Durch einen Klick wird der Benutzer zur Seite "Rettungskarte" des entsprechenden Fahrzeuges weitergeleitet.

Der Benutzer hat zudem die Möglichkeit, die Zoom-Funktion in der linken unteren Ecke der Karte zu verwenden, um das Fahrzeug und seinen Zustand detaillierter zu betrachten. Die Auswirkungen des Unfalls werden simuliert, um die Effekte unterschiedlicher Kräfte und Geschwindigkeiten zu veranschaulichen.

Um zur Suchseite zurückzukehren, kann der Benutzer auf das Home-Symbol in der oberen rechten Ecke klicken. Die Webanwendung ist dann für eine erneute Suche bereit.

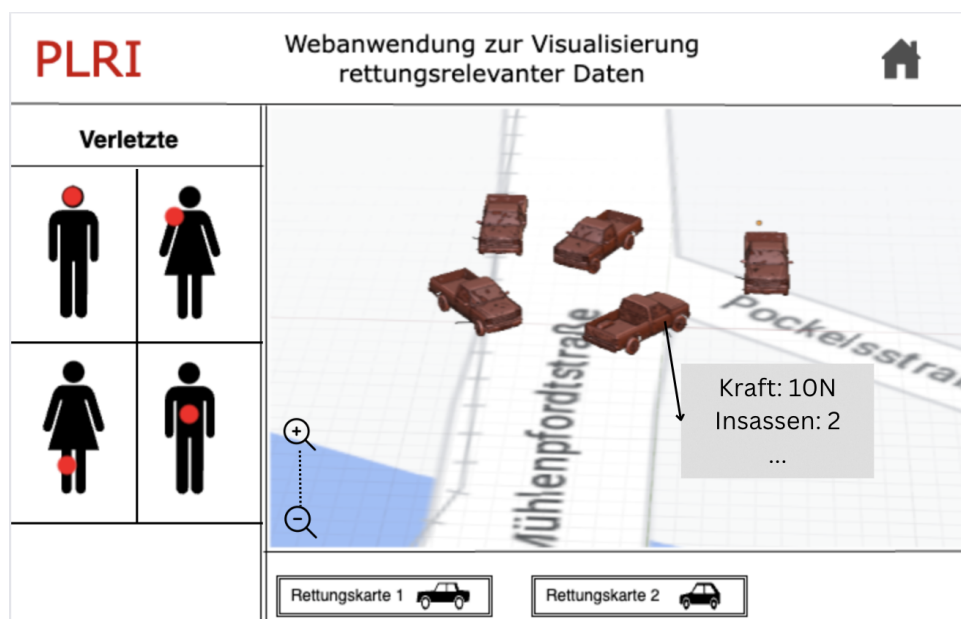


Abbildung 7.2: Karte $\langle UI20 \rangle$

Rettungskarte $\langle UI30 \rangle$

Auf der Rettungskarte werden die wichtigsten Informationen über ein Fahrzeug in einer logischen Reihenfolge dargestellt. In der Regel enthält die Rettungskarte Details wie Gewicht, Abmessungen und Karosseriematerialien des Fahrzeugs sowie die Position von Kraftstofftanks, Batterien und Airbags. Die meisten dieser Informationen werden auch in einer 3D-Ansicht des Fahrzeuges in der Rettungskarte angezeigt. Um zur Kartenansicht zurückzukehren, kann der Benutzer

einfach auf den Pfeil in der oberen linken Ecke der Seite klicken.

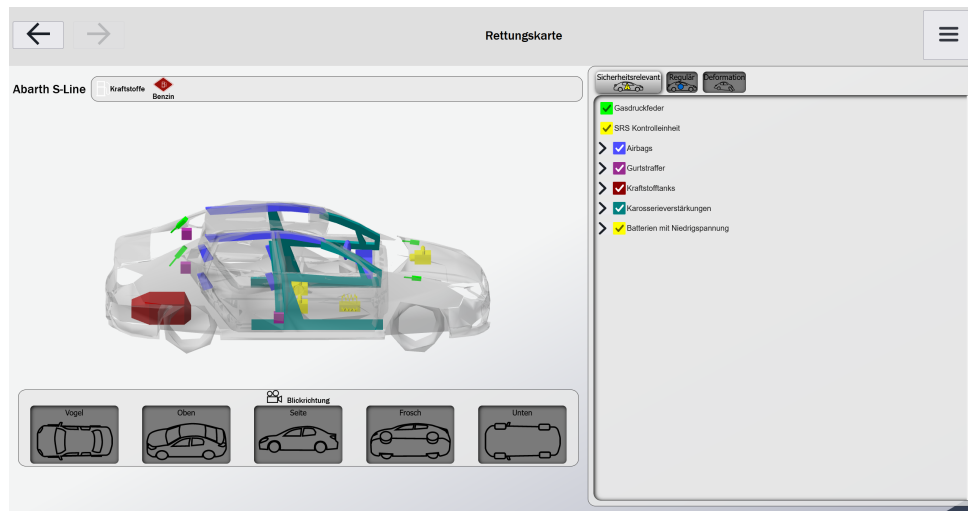


Abbildung 7.3: Suche $\langle UI30 \rangle$

8 Technische Produktumgebung

In diesem Kapitel wird die technische Umgebung des Produktes beschrieben. Bei Client/Server-Anwendungen ist die Umgebung jeweils für Client und Server getrennt anzugeben.

8.1 Software

- Server-Betriebssystem: Windows 10 64-Bit
- Client-Betriebssystem: Web-browser
- Implementierungssprache der Webanwendung: HTML, Typescript
- Implementierungssprache des Simulations-Anwendung: Python (Flask)
- Simulationsumgebung: BeamNG (Python, Lua)
- Front-End-Webapplikationsframework: Angular

8.2 Hardware

- Server: Virtuellen Server, der die Mindestanforderungen für den Betrieb von BeamNG erfüllt mit 8-Core CPU, 16 GB RAM, 8 GB Grafikkarte, und 45 GB Speicherplatz.
- Client: Standard PC oder ein browserfähiges Gerät mit Grafikbildschirm z.B. NidaPad.

8.3 Produktschnittstellen

Die Benutzerschnittstelle wird über eine Webanwendung dargestellt. Die Kommunikation zwischen dem Server und der Webseite erfolgt mit Hilfe von REST APIs.

9 Glossar

Angular ist ein TypeScript-basiertes Front-End-Webapplikationsframework. Es wird von einer Community aus Einzelpersonen und Unternehmen, angeführt durch Google, entwickelt und als Open-Source-Software publiziert.

BeamNGpy ist eine offizielle Bibliothek, die eine Python-API für BeamNG.tech bereitstellt, den akademie- und industrieorientierten Fork des Videospiels BeamNG.drive.

Frontend und **Backend** werden in der Informationstechnik an verschiedenen Stellen in Verbindung mit einer Schichteneinteilung verwendet. Dabei ist typischerweise das Frontend näher am Benutzer, das Backend näher am System.

HSN und TSN ergeben zusammen einen alphanumerischen Code, der nicht nur Zahlen, sondern auch Buchstaben enthält. Die Kombination aus HSN und TSN gibt unter anderem Auskunft über den Fahrzeugtyp, die Motorleistung, den Hubraum und die Art des Kraftstoffes. **HTML** ist eine textbasierte Auszeichnungssprache zur Strukturierung elektronischer Dokumente wie Texte mit Hyperlinks, Bildern und anderen Inhalten.

ISAN (International Standard Accident Number) Das ISAN-Projekt zielt darauf ab große Datenmengen aus diversen Quellen, wie der Notfallmedizin (EMS), der elektronischen Gesundheitsakte (EHR) und Ereignisdatenschreibern (EDR) zu sammeln und durch die Schaffung einer technischen Grundlage zu verbinden und Rettungseinsätze zu unterstützen. Die ISAN selbst enthält Unfalldaten.

Klasse in der objektorientierten Programmierung ist ein abstraktes Modell bzw. einen Bauplan für eine Reihe von ähnlichen Objekten.

NidaPad ist ein Tabletcomputer, der gezielt für Rettungskräfte und Einsatzbedingungen entwickelt wurde. Das NIDApad unterstützt die Kommunikation zwischen Leitstelle und Rettungswagen, die Navigation zum Einsatzort, eine umfassende Einsatzdokumentation sowie die Weitergabe der Einsatzdaten an die Klinik.

Python ist eine einfach zu erlernende, interpretierte, objektorientierte Programmiersprache mit

dynamischer Typisierung.

Rettungskarte ist eine bereits vorhandene Softwarekomponente, die Informationen eines spezifischen Fahrzeugs enthält, welche den Rettungskräften helfen, das Fahrzeug gefahrlos zu öffnen.

Texture/Vertex Heatmap ist ein Diagrammtyp zur farblichen Visualisierung von Daten im zweidimensionalen Raum/ einer Ebene.

TypeScript ist eine kostenlose und Open-Source-Programmiersprache, die auf JavaScript aufbaut und statische Typisierung unterstützt.

3D-Mesh ist eine dreidimensionale Struktur, die ein 3D-Objekt darstellt.