



WEBANWENDUNG ZUR VISUALISIERUNG VON RETTUNGSRELEVANTEN DATEN ZUR EINSATZUNTERSTÜTZUNG

Software-Entwicklungspraktikum (SEP)
Sommersemester 2023

Testprotokolle

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, 12. Juli 2023

Bearbeiterübersicht

Kapitel	Autoren	Kommentare
1	Jonas Stepanik	keine
1.1	Jonas Stepanik	keine
1.2	Jonas Stepanik	keine
1.3	Jonas Stepanik	keine
2	Mohmaed Wassim Che- bili	keine
2.1	Mohmaed Wassim Che- bili	keine
2.2	Mohmaed Wassim Che- bili	keine
2.3	Mohmaed Wassim Che- bili	keine
3	Omar Farouk Khayat	keine
3.1	Omar Farouk Khayat	keine
3.2	Omar Farouk Khayat	keine
3.3	Omar Farouk Khayat	keine
4	Torben Oelerking	keine
4.1	Torben Oelerking	keine
4.2	Torben Oelerking	keine
4.3	Torben Oelerking	keine
5	Omar Farouk Khayat	keine
5.1	Omar Farouk Khayat	keine
5.2	Omar Farouk Khayat	keine
5.3	Omar Farouk Khayat	keine
6	Mohmaed Wassim Che- bili	keine
6.1	Mohmaed Wassim Che- bili	keine
6.2	Mohmaed Wassim Che- bili	keine
6.3	Mohmaed Wassim Che- bili	keine
7	Kacem Abdennabih	keine
7.1	Kacem Abdennabih	keine

7.2	Kacem Abdennabih	keine
7.3	Kacem Abdennabih	keine
8	Torben Oelerking	keine
8.1	Torben Oelerking	keine
8.2	Torben Oelerking	keine
8.3	Torben Oelerking	keine
9	Omar Farouk Khayat	keine
9.1	Omar Farouk Khayat	keine
9.2	Omar Farouk Khayat	keine
9.3	Omar Farouk Khayat	keine
10	Kacem Abdennabih	keine
10.1	Kacem Abdennabih	keine
10.2	Kacem Abdennabih	keine
10.3	Kacem Abdennabih	keine
11	Omar Farouk Khayat	keine
11.1	Omar Farouk Khayat	keine
11.2	Omar Farouk Khayat	keine
11.3	Omar Farouk Khayat	keine
12	Kacem Abdennabih	keine
12.1	Kacem Abdennabih	keine
12.2	Kacem Abdennabih	keine
12.3	Kacem Abdennabih	keine
13	Azhar Rahadian	keine
13.1	Azhar Rahadian	keine
13.2	Azhar Rahadian	keine
13.3	Azhar Rahadian	keine
14	Mohamed Wassim Che- bili	keine
14.1	Mohamed Wassim Che- bili	keine
14.2	Mohamed Wassim Che- bili	keine
14.3	Mohamed Wassim Che- bili	keine
15	Qiyue Zhang	keine
15.1	Qiyue Zhang	keine
15.2	Qiyue Zhang	keine

15.3	Qiyue Zhang	keine
------	-------------	-------

Inhaltsverzeichnis

1	Testdurchführung (2023-07-09)	7
1.1	Testumgebung	7
1.2	Testprotokoll	7
1.3	Zusammenfassung	8
2	Testdurchführung (2023-07-10)	9
2.1	Testumgebung	9
2.2	Testprotokoll	9
2.3	Zusammenfassung	10
3	Testdurchführung (2023-07-10)	11
3.1	Testumgebung	11
3.2	Testprotokoll	11
3.3	Zusammenfassung	12
4	Testdurchführung (2023-07-11))	13
4.1	Testumgebung	13
4.2	Testprotokoll	13
4.3	Zusammenfassung	14
5	Testdurchführung (2023-07-11)	15
5.1	Testumgebung	15
5.2	Testprotokoll	15
5.3	Zusammenfassung	16
6	Testdurchführung (2023-07-11)	17
6.1	Testumgebung	17
6.2	Testprotokoll	17
6.3	Zusammenfassung	17
7	Testdurchführung (2023-07-12)	19
7.1	Testumgebung	19
7.2	Testprotokoll	19
7.3	Zusammenfassung	19

8 Testdurchführung (2023-07-12)	20
8.1 Testumgebung	20
8.2 Testprotokoll	20
8.3 Zusammenfassung	21
9 Testdurchführung (2023-07-12)	22
9.1 Testumgebung	22
9.2 Testprotokoll	22
9.3 Zusammenfassung	23
10 Testdurchführung (2023-07-12)	24
10.1 Testumgebung	24
10.2 Testprotokoll	24
10.3 Zusammenfassung	25
11 Testdurchführung (2023-07-12)	26
11.1 Testumgebung	26
11.2 Testprotokoll	26
11.3 Zusammenfassung	26
12 Testdurchführung (2023-07-12)	27
12.1 Testumgebung	27
12.2 Testprotokoll	27
12.3 Zusammenfassung	28
13 Testdurchführung (2023-07-10)	29
13.1 Testumgebung	29
13.2 Testprotokoll	29
13.3 Zusammenfassung	30
14 Testdurchführung (2023-07-12)	31
14.1 Testumgebung	31
14.2 Testprotokoll	31
14.3 Zusammenfassung	31
15 Testdurchführung (2023-07-12)	32
15.1 Testumgebung	32
15.2 Testprotokoll	32
15.3 Zusammenfassung	32

1 Testdurchführung (2023-07-09)

Dieser Test befasst sich mit der Funktion **F3** und validiert somit die korrekte Arbeitsweise des gesamten Simulations-Servers. Dazu werden die zwei groben Unterkomponenten des Simulations-Servers getestet: Die Simulation und die REST-API.

Art des Tests: Unit-Test

Ausgeführte Testfälle: **T20 T5**

Beteiligte Tester: Jonas Stepanik

Abgedeckte Funktionen: **F3**

1.1 Testumgebung

Die Tests werden unter Windows 10 durchgeführt. Die BeamNG.tech Version ist 0.28.1.0. Die Versionen der Python-Module können im Repository nachgeschaut werden.

1.2 Testprotokoll

Die folgenden Tabellen beschreiben, wie der Testfall ausgeführt wurde und welches Ergebnis er geliefert hat. Da es bei Korrektur von Softwarefehlern oder anderen Gegebenheiten notwendig ist, einen Test mehrfach durchzuführen (Testläufe), ist jede Testdurchführung zu dokumentieren. Daher ist diese Tabelle für **jeden Testlauf** zu erstellen und **fortlaufend zu nummerieren**.

Testfall	T20
Tester	Jonas Stepanik
Eingaben	Der Test läuft vollautomatisch, indem man die Datei 'plri_0/Simulation/python/test_simulator.py' ausführt.
Soll - Reaktion	Wenn der Test erfolgreich ist, wird 'OK' ausgegeben.
Ist – Reaktion	Der Test hat 'OK' ausgegeben.
Ergebnis	Der Test war erfolgreich.

Testfall	T5
Tester	Jonas Stepanik
Eingaben speed=[FILL] tsn=[FILL]'	Wenn der REST-Server auf dem localhost läuft führen wir 5 curl-commands aus, um Anfragen zu stellen. Die curl-commands haben folgende Form: `curl -X GET 'http://128.0.0.1:5000/simulation?force=[FILL] hsn=[FILL]` Wir übergeben die Parameter: 1. force=50, speed=50, hsn=0588, tsn=637 2. force=100, speed=100, hsn=0005, tsn=BSN 3. force=150, speed=150, hsn=7966, tsn=AFS 4. force=200, speed=200, hsn=0005, tsn=BSN 5. force=10, speed=10, hsn=4192, tsn=ADY
Soll - Reaktion	Wenn der Test erfolgreich ist, gibt jede Anfrage eine erfolgreiche Response (Code 200) und das Mesh zurück.
Ist – Reaktion	Alle Anfragen haben einen erfolgreichen Response-Code und ein Mesh zurückgeliefert.
Ergebnis	Der Test war erfolgreich.

1.3 Zusammenfassung

Da beide Tests ohne weitere Probleme erfolgreich sind, ist sichergestellt, dass Funktion **F3** ordnungsgemäß funktioniert. Es sind keine Abweichungen von zuvor vereinbarten Aspekten der Funktion zu verzeichnen. Die Qualität der Software ist sehr hoch, denn es wurde durchgehend kommentiert, die Formatierung ist einheitlich und in sämtlichen Modulen ist Logging implementiert, sodass stets die Log-Datei geprüft werden kann, falls ein Fehler auftritt.

2 Testdurchführung (2023-07-10)

Art des Tests: Unit-Test

Ausgeführte Testfälle: **T22 T23 T24**

Beteiligte Tester: Mohamed Wassim Chebili

Abgedeckte Funktionen: **F5**

2.1 Testumgebung

Die Tests werden auf einem Windows 10-System ausgeführt, unter Verwendung von Google Chrome Version 114.0.5735.199. Die Testskripte wurden mit Angular Version 1.8.3, Karma Version 6.4.2 und Jasmine Version 5.0.2 entwickelt.

2.2 Testprotokoll

Testfall	T22
Tester	Mohamed Wassim Chebili
Eingaben	Im kommandozeile “ng test --include=src/app/accident.service.spec.ts”
Soll - Reaktion	Der Test 'should get accident by id' soll im Browser nicht als fehlgeschlagen angezeigt werden.
Ist – Reaktion	Der Test 'should get accident by id' wurde im Browser nicht als fehlgeschlagen angezeigt.
Ergebnis	Der Test war erfolgreich.

Testfall	T23
Tester	Mohamed Wassim Chebili
Eingaben	Im kommandozeile “ng test --include=src/app/accident.service.spec.ts”
Soll - Reaktion	Der Test 'should get cached accident' soll im Browser nicht als fehlgeschlagen angezeigt werden.

Ist – Reaktion	Der Test 'should get cached accident' wurde im Browser nicht als fehlgeschlagen angezeigt.
Ergebnis	Der Test war erfolgreich.

Testfall	T24
Tester	Mohamed Wassim Chebili
Eingaben	Im kommandozeile “ng test –include=src/app/accident.service.spec.ts” ausführen
Soll - Reaktion	Der Test 'should reset accident' soll im Browser nicht als fehlgeschlagen angezeigt werden.
Ist – Reaktion	Der Test 'should reset accident' wurde im Browser nicht als fehlgeschlagen angezeigt.
Ergebnis	Der Test war erfolgreich.

2.3 Zusammenfassung

- Alle drei Tests waren erfolgreich, was bedeutet, dass die Funktionalität des Unfall-Services gewährleistet ist. Der Unfall-Service kann einen Unfall ordnungsgemäß anfragen, zwischenspeichern und anschließend korrekt löschen.
- Es gab keine Abweichungen von der Aufgabenstellung oder dem Testplan.
- Der Test war vollständig.
- Die Softwarequalität ist gut.

3 Testdurchführung (2023-07-10)

Art des Tests: Unit-Test

Ausgeführte Testfälle: **T25 T26**

Beteiligte Tester: Omar Farouk Khayat

Abgedeckte Funktionen: **F4, F5, F7**

3.1 Testumgebung

Es wurden Testskripte entwickelt, die Angular Version 1.8.3, Karma Version 6.4.2 und Jasmine Version 5.0.2 verwenden. Diese Tests werden auf einem Windows 10-System ausgeführt und verwenden Google Chrome Version 114.0.5735.199(64-Bit).

3.2 Testprotokoll

Testfall	T25
Tester	Omar Farouk Khayat
Eingaben	Der Test läuft vollautomatisch, indem man die Datei <code>ng test --include=src/app/accident-page/accident-page.component.spec.ts</code> ausführt.
Soll - Reaktion	Der Test should load accident data on component initialization soll im Browser als erfolgreich angezeigt werden.
Ist – Reaktion	Der Test should load accident data on component initialization "wurde im Browser als erfolgreich angezeigt.
Ergebnis	Der Test war erfolgreich.

Testfall	T26
Tester	Omar Farouk Khayat
Eingaben	Der Test läuft vollautomatisch, indem man die Datei <code>ng test --include=src/app/home-page/home-page.component.spec.ts</code> ausführt

Soll - Reaktion	Der Test should go to new page when input is given"im Browser als erfolgreich angezeigt werden.
Ist – Reaktion	Der Test should go to new page when input is given"wurde im Browser als erfolgreich angezeigt.
Ergebnis	Der Test war erfolgreich.

3.3 Zusammenfassung

Beide Tests waren erfolgreich, was bedeutet, dass der Nutzer eine Unfall-ID eingeben kann und dadurch erfolgreich einen Unfall laden und zur Home-Page weitergeleitet werden kann. Die Accident-Page ist ebenfalls in der Lage, den Unfall korrekt vom Unfallservice anzufordern.

Ist – Reaktion	Die Webanwendung lädt die Google-Maps Karte. Auf dieser Karte werden die zwei Unfallfahrzeuge angezeigt. An der Position eines jeden Unfallfahrzeuges wird ein Info Marker angezeigt. Bei Auswahl eines Marker, werden Zusatzinformationen angezeigt. Das Infofenster lässt sich schließen. Mit Hilfe der Navigationstasten auf der Karte kann um den Unfall im 3D-Raum navigiert werden. In der Console im Browser ist sichtbar, dass "Mesh 0" und "Mesh 1" geladen wurden.
Ergebnis	Der Test war erfolgreich.

4.3 Zusammenfassung

Der Test war erfolgreich, was bedeutet, dass der Nutzer eine Unfall-ID eingeben kann und dadurch erfolgreich einen Unfall laden und zur Home-Page weitergeleitet werden kann. Die Accident-Page ist ebenfalls in der Lage, den Unfall korrekt vom Unfallservice anzufordern.

5 Testdurchführung (2023-07-11)

Art des Tests: Unit-Test

Ausgeführte Testfälle: **T31**

Beteiligte Tester: Kacem Abdennabih

Abgedeckte Funktionen: **F4**, **F8**

5.1 Testumgebung

Die Testumgebung ist Jest, ein beliebtes JavaScript-Testframework, das häufig zum Testen von Node.js-Anwendungen verwendet wird.

5.2 Testprotokoll

Testfall	T31
Tester	Kacem Abdennabih
Eingaben Mocked file path: /mock/path/to/accident_23.json Mocked accident data: id:123, location: Example Location	id: 123
Soll - Reaktion	Die Webanwendung lädt die Google-Maps Karte. Auf dieser Karte werden die zwei Unfallfahrzeuge angezeigt. An der Position eines jeden Unfallfahrzeuges wird ein Info Marker angezeigt. Bei Auswahl eines Marker, werden Zusatzinformationen angezeigt. Das Infofenster lässt sich schließen. Mit Hilfe der Navigationstasten auf der Karte kann um den Unfall im 3D-Raum navigiert werden. In der Console im Browser ist sichtbar, dass "Mesh 0" und "Mesh 1" geladen wurden.

Ist – Reaktion	Die Webanwendung lädt die Google-Maps Karte. Auf dieser Karte werden die zwei Unfallfahrzeuge angezeigt. An der Position eines jeden Unfallfahrzeuges wird ein Info Marker angezeigt. Bei Auswahl eines Marker, werden Zusatzinformationen angezeigt. Das Infofenster lässt sich schließen. Mit Hilfe der Navigationstasten auf der Karte kann um den Unfall im 3D-Raum navigiert werden. In der Console im Browser ist sichtbar, dass "Mesh 0" und "Mesh 1" geladen wurden.
Ergebnis	Der Test war erfolgreich.

5.3 Zusammenfassung

Der Test war erfolgreich, was bedeutet, dass der Nutzer eine Unfall-ID eingeben kann und dadurch erfolgreich einen Unfall laden und zur Home-Page weitergeleitet werden kann. Die Accident-Page ist ebenfalls in der Lage, den Unfall korrekt vom Unfallservice anzufordern.

6 Testdurchführung (2023-07-11)

Art des Tests: Integrations-Test

Ausgeführte Testfälle: **T13** Beteiligte Tester: Omar Farouk Khayat

Abgedeckte Funktionen: **F5**

6.1 Testumgebung

Es wurden Testskripte entwickelt, die Angular Version 1.8.3, Karma Version 6.4.2 und Jasmine Version 5.0.2 verwenden. Diese Tests werden auf einem Windows 10-System ausgeführt und verwenden Google Chrome Version 114.0.5735.199(64-Bit).

6.2 Testprotokoll

Testfall	T13
Tester	Omar Farouk Khayat
Eingaben	Der Test läuft vollautomatisch, indem man die Datei <code>'ng test --sinclude=src/app/accident-page/accident-page.component.integration.spec.ts'</code> ausführt.
Soll - Reaktion	Der Test "getAccidentByID" soll im Browser als erfolgreich angezeigt werden.
Ist – Reaktion	Der Test "getAccidentByID" wurde im Browser als erfolgreich angezeigt.
Ergebnis	Der Test war erfolgreich.

6.3 Zusammenfassung

- Der Test war erfolgreich, was bedeutet, dass der Unfallservice die empfangenen Daten vom Webserver korrekt verarbeitet und speichert. Dadurch wird sichergestellt, dass die Daten erfolgreich übertragen und vom Unfallserver richtig verarbeitet werden. .

- Es gab keine Abweichungen von der Aufgabenstellung oder dem Testplan.
- Der Test war vollständig.
- Die Softwarequalität ist gut.

7 Testdurchführung (2023-07-12)

Art des Tests: Unit-Test

Ausgeführte Testfälle: ?? Beteiligte Tester: Kacem Abdennabih

Abgedeckte Funktionen: F5

7.1 Testumgebung

Die Testumgebung ist Jest, ein beliebtes JavaScript-Testframework, das häufig zum Testen von Node.js-Anwendungen verwendet wird.

7.2 Testprotokoll

Testfall	27
Tester	Kacem Abdennabih
Eingaben	id:123, Mocked file path: /mock/path/to/accident123.json, Mocked accident data : <i>id : '123', location : 'ExampleLocation'</i>
Soll - Reaktion	Die Funktion 'fs.readFile' sollte mit 'mockFilePath', 'utf8' und einer anonymen Callback-Funktion aufgerufen werden.
Ist – Reaktion	Der Test wurde bestanden, was bedeutet, dass die erwartete Ausgabe mit dem tatsächlichen Verhalten der Funktion übereinstimmt.
Ergebnis	Der Test war erfolgreich.

7.3 Zusammenfassung

Allgemein wurden alle Testfälle bestanden, was darauf hindeutet, dass sich die Funktion "getAccidentById" in verschiedenen Szenarien wie erwartet verhält.

8 Testdurchführung (2023-07-12)

Art des Tests: Unit-Test

Ausgeführte Testfälle: ?? Beteiligte Tester: Kacem Abdennabih

Abgedeckte Funktionen: **F6**

8.1 Testumgebung

Die Testumgebung ist Jest, ein beliebtes JavaScript-Testframework, das häufig zum Testen von Node.js-Anwendungen verwendet wird.

8.2 Testprotokoll

Testfall	28
Tester	Kacem Abdennabih
Eingaben	Es existieren drei Unfalldateien im ./mockData/accidents Ordner. Jede der drei Unfalldateien hat ein abgelaufenes Datum und eine abgelaufene Uhrzeit
Soll - Reaktion	Jede der drei Unfalldateien soll erfolgreich gelöscht werden. Es sollen entsprechende Erfolgsmeldungen für jede gelöschte Datei ausgegeben werden
Ist – Reaktion	Die fs.readdir Funktion wird mit dem richtigen Ordnerpfad aufgerufen. Die fs.readFile Funktion wird dreimal aufgerufen, einmal für jede Unfalldatei. Die fs.unlink Funktion wird dreimal aufgerufen, einmal für jede abgelaufene Unfalldatei. Die entsprechenden Erfolgsmeldungen für jede gelöschte Datei werden korrekt ausgegeben
Ergebnis	Der Test war erfolgreich.

8.3 Zusammenfassung

Alle abgelaufenen Unfalldateien werden erfolgreich gelöscht. Die Funktion verhält sich wie erwartet und gibt die entsprechenden Erfolgsmeldungen aus.

9 Testdurchführung (2023-07-12)

Art des Tests: Abnahmetest

Ausgeführte Testfälle: **T6T10**

Beteiligte Tester: Torben Oelerking

Abgedeckte Funktionen: **F4F8**

9.1 Testumgebung

Es werden keine Testskripte verwendet. Die Angular Version lautet 1.8.3. Diese Tests werden auf einem Ubuntu 64-bit System ausgeführt und unter Verwendung der Firefox Version 115.0.2 (64-bit).

9.2 Testprotokoll

Testfall	??
Tester	Torben Oelerking
Eingaben	Starten der Anwendung mittel "ng serveüind "json-server –watch db.jsonEingabe der Unfall-ID 22 in die Suche.
Soll - Reaktion	Der Test gilt als erfolgreich, wenn die Webanwendung die Daten eines Unfalls vom Webserver empfangen kann und den Unfall entsprechend korrekt in der Webanwendung (3D-Modelle der Fahrzeuge werden auf der Karte angezeigt) darstellt, wobei die angezeigten Informationen bei Auswahl des entsprechenden Infosymbols den gegebenen Daten entsprechen.
Ist – Reaktion	Die Webanwendung empfängt die Daten eines Unfalls vom Webserver und stellt den Unfall korrekt in der Webanwendung dar(3D-Modelle der Fahrzeuge werden auf der Karte angezeigt) wobei die angezeigten Informationen bei Auswahl des entsprechenden Infosymbols den gegebenen Daten entsprechen.
Ergebnis	Der Test war erfolgreich.

Testfall	??
Tester	Torben Oelerking
Eingaben	Starten der Anwendung mittel "ng serveüind "json-server –watch db.json Eingabe der Unfall-ID 22 in die Suche.
Soll - Reaktion	Der Test gilt als bestanden, falls sich das Fenster korrekt öffnet und schließt und es auch die korrekten Informationen anzeigt.
Ist – Reaktion	Das Fenster öffnet und schließt sich korrekt und es werden auch die korrekten Informationen anzeigt.
Ergebnis	Der Test war erfolgreich.

9.3 Zusammenfassung

- Die beiden Tests waren erfolgreich, was bedeutet, dass die Funktionalität der Visualisierung der Unfälle auf der Karte und das Anzeigen der Zusatzinformationen gewährleistet ist.
- Das Ziel des Testziels hat sich dahingehend geändert, dass das Anzeigen der Informationen nicht durch Hoovern des Mauszeigers über dem entsprechenden Unfallfahrzeug sondern durch das Anklicken des Infosymbols an der Stelle des Fahrzeuges erfolgt.
- Der Test war vollständig.
- Die Softwarequalität ist gut.

10 Testdurchführung (2023-07-12)

Art des Tests: Abnahmetest

Ausgeführte Testfälle: **T7 T9** Beteiligte Tester: Omar Farouk Khayat

Abgedeckte Funktionen: **F5, F7**

10.1 Testumgebung

Es wurden Tests entwickelt, die die Angular Version 1.8.3, Karma Version 6.4.2 und Jasmine Version 5.0.2 verwenden. Diese Tests werden auf einem Windows 10-System ausgeführt und nutzen Google Chrome Version 114.0.5735.199 (64-Bit).

10.2 Testprotokoll

Testfall	T7
Tester	Omar Farouk Khayat
Eingaben	1.Fall: Es wird die ID 22 eingegeben, die zum entsprechenden Unfall gehört 2.Fall:Es wird eine 23 eingegeben, die nicht zum entsprechenden Unfall gehört.
Soll - Reaktion	Wenn der Benutzer die ID 22 eingibt, wird die Unfallseite angezeigt. Wenn die ID 23 eingegeben wird, erscheint die Meldung "No Accident Found".
Ist – Reaktion	Der Benutzer gibt die ID 22 ein und die Unfallseite wird angezeigt.Der Benutzer gibt die ID 23 ein und die Meldung "No Accident Found"wird angezeigt.
Ergebnis	Der Test war erfolgreich.

Testfall	T9
Tester	Omar Farouk Khayat
Eingaben	Es wird die ID 22 eingegeben, die zum entsprechenden Unfall gehört, und anschließend wird der "RettungskarteButton gedrückt.

Soll - Reaktion	Wenn der Benutzer die ID 22 eingibt, wird die Unfallseite angezeigt. Wenn er das "RettungskarteButton drückt, erscheint die Rettungskarte.
Ist – Reaktion	Der Benutzer gibt die ID 22 ein und die Unfallseite wird angezeigt. Der Benutzer drückt "RettungskarteButton und Die Rettungskarte wird angezeigt.
Ergebnis	Der Test war erfolgreich.

10.3 Zusammenfassung

- Beide Tests waren erfolgreich, was darauf hindeutet, dass die Funktionen zur Suche nach einem Unfall und zum Aufrufen der Rettungskarte einwandfrei funktionieren. Sowohl die Suche nach einem Unfall als auch das Aufrufen der entsprechenden Rettungskarte wurden korrekt durchgeführt und die richtigen Ergebnisse mit den entsprechenden IDs wurden angezeigt.
- Es gab keine Abweichungen von der Aufgabenstellung oder dem Testplan.
- Der Test war vollständig.
- Die Softwarequalität ist gut.

11 Testdurchführung (2023-07-12)

Art des Tests: Abnahmetest

Ausgeführte Testfälle: **T8** Beteiligte Tester: Kacem Abdennabih

Abgedeckte Funktionen: **F6**

11.1 Testumgebung

Es wurden Test entwickelt, die Angular Version 1.8.3, Karma Version 6.4.2 und Jasmine Version 5.0.2 verwenden. Sie werden auf einem Windows 10-System ausgeführt und verwenden Google Chrome Version 114.0.5735.199(64-Bit).

11.2 Testprotokoll

Testfall	T8
Tester	Kacem Abdennabih
Eingaben	Der Zeitraum für das Löschen der Unfalldaten wurde von 5 Stunden auf 5 Minuten reduziert, um die Tests zu vereinfachen und die Testdurchführung zu beschleunigen. Es wird danach mit Postman Diese Request geschickt "ISAN": "+01052037;+5227741;113026;20230601;30;60;0603;ANY;BSAO123
Soll - Reaktion	Der Unfall soll nach 5 Minuten gelöscht werden
Ist – Reaktion	Der Unfall ist nach 5 Minuten gelöscht.
Ergebnis	Der Test war erfolgreich.

11.3 Zusammenfassung

- Die Funktion löscht abgelaufene Unfalldateien erfolgreich und gibt die erwarteten Erfolgsmeldungen aus
- Der Test war vollständig.

12 Testdurchführung (2023-07-12)

Art des Tests: Abnahmetest

Ausgeführte Testfälle: **T1 T2**

Beteiligte Tester: Azhar Rahadian

Abgedeckte Funktionen: **F1**

12.1 Testumgebung

Die Tests werden auf einem Windows 10-System ausgeführt, unter Verwendung von Google Chrome Version 114.0.5735.199. Die Testskripte wurden mit Angular Version 1.8.3, Karma Version 6.4.2 und Jasmine Version 5.0.2 entwickelt.

12.2 Testprotokoll

Testfall	T1
Tester	Azhar Rahadian
Eingaben	Postman wird verwendet um die POST Request zu erstellen. ISAN mit korrektem Format wird übergeben 'ISAN':'+01052037;+5227741;113026;20230601;30;60;0603;ANY;BSAO123;4'
Soll - Reaktion	Der Test ist erfolgreich, wenn die Anwendung ein erfolgreichen Response (Code 201) gibt, die korrekten Daten aus einer gültigen ISAN extrahiert und erfolgreich einen neuen Unfall auf dem Server speichert.
Ist – Reaktion	Der Test gibt den erfolgreichen Response zurück, hat die Daten aus der ISAN extrahiert und einen neuen Unfall wurde auf dem Server gespeichert.
Ergebnis	Der Test war erfolgreich.

Testfall	T2
Tester	Azhar Rahadian

Eingaben	Postman wird verwendet um die POST Request zu erstellen. ISAN mit falschem Format wird übergeben "ISAN":"+0105;+5227741;113026;20230601;30;60;0603;ANY;BSAO123;4"
Soll - Reaktion	Der Test ist erfolgreich, ob die Anwendung eine ISAN mit falschem Format nicht akzeptiert bzw. entsprechend reagiert (Code 403).
Ist – Reaktion	Der Test hat die ISAN mit falschem Format nicht akzeptiert und eine nicht erfolgreichen Response wird zurückgegeben
Ergebnis	Der Test war erfolgreich.

12.3 Zusammenfassung

Alle zwei Teste wurden erfolgreich getestet. Die Funktionalität von Auswertung einer ISAN wird sichergestellt. Es gab keine Abweichungen von der Aufgabenstellung oder dem Testplan und der Test war vollständig. Die Softwarequalität ist hoch, denn die Auswertung der ISAN Format erfolgt mit strenger Kontrolle damit es keine falsche Unfälle hinzugefügt werden können, und am ende wird eine Response von jeweilige Fälle der Auswertung zurückgegeben.

13 Testdurchführung (2023-07-10)

Art des Tests: Abnahmetest

Ausgeführte Testfälle: **T3 T4**

Beteiligte Tester: Mohamed Wassim Chebili

Abgedeckte Funktionen: **F2**

13.1 Testumgebung

Die Tests werden auf einem Windows 10-System ausgeführt und mit Node.js v18.16.0.

13.2 Testprotokoll

Testfall	??
Tester	Mohamed Wassim Chebili
Eingaben	Wir nutzen Postman um die Post Request zu erstellen. Es werden diese Requests geschickt: İSAN":"+01052037;+5227741;113026;20230601;30;60;0603;ANY;BSAO123;4" İSAN":"+01052034;+5227739;113025;20230601;30;60;0603;ANY;BSAO123;4" İSAN":"+01052034;+5227740;113027;20230601;30;60;0603;ANY;BSAO123;4"
Soll - Reaktion	Es soll eine Unfalldatei erstellt werden, die alle 3 Unfälle enthält. Diese Datei kann dann überprüft werden.
Ist – Reaktion	Eine Unfalldatei mit allen drei Unfällen wurde erstellt. Die Daten der Datei wurden überprüft und stimmen mit den Anfragen überein
Ergebnis	Der Test war erfolgreich.

Testfall	??
Tester	Mohamed Wassim Chebili

Eingaben	Wir nutzen Postman um die Post Request zu erstellen. Es werden diese Requests geschickt: İSAN":"+01052037;+5227741;113026;20230601;30;60;0603;ANY;BSAO123;4" İSAN":"+01152034;+5227739;113025;20230601;30;60;0603;ANY;BSAO123;4" İSAN":"+01052034;+5227740;113225;20230601;30;60;0603;ANY;BSAO123;4"
Soll - Reaktion	Alle 3 Unfälle sollen ihre eigene Datei haben. Der Inhalt der Datei sollte überprüft werden.//
Ist – Reaktion	Alle 3 Unfälle wurden mit ihrer eigenen Datei erstellt. Der Inhalt der Datei stimmt mit den ISANs überein.//
Ergebnis	Der Test war erfolgreich.

13.3 Zusammenfassung

- Die beiden Tests waren erfolgreich, was bedeutet, dass unser Webserver in der Lage ist, Unfälle zu erkennen und die zugehörigen Daten zu gruppieren oder in einer separaten Datei zu speichern.
- Es gab keine Abweichungen von der Aufgabenstellung oder dem Testplan.
- Der Test war vollständig.
- Die Softwarequalität ist gut.

14 Testdurchführung (2023-07-12)

Art des Tests: Unit-Test

Ausgeführte Testfälle: ?? Beteiligte Tester: Qiyue Zhang

Abgedeckte Funktionen: F1

14.1 Testumgebung

Die Testumgebung ist Jest, ein beliebtes JavaScript-Testframework, das häufig zum Testen von Node.js-Anwendungen verwendet wird.

14.2 Testprotokoll

Testfall	29
Tester	Qiyue Zhang
Eingaben	Wir nutzen Postman um die Post Request zu erstellen. Es werden diese Requests geschickt: İSAN":"+01052037;+5227741;113026;20230601;30;60;0603;ANY;BSAO123;4" İSAN":"+01052034;+5227739;113025;20230601;30;60;0603;ANY;BSAO123;4" İSAN":"+01052034;+5227740;113027;20230601;30;60;0603;ANY;BSAO123;4"
Soll - Reaktion	Der Statuscode soll 201 sein und die Erfolgsmeldung soll zurückgegeben werden.
Ist – Reaktion	Der Test wurde bestanden, was bedeutet, dass die erwartete Ausgabe mit dem tatsächlichen Verhalten der Funktion übereinstimmt.
Ergebnis	Der Test war erfolgreich.

14.3 Zusammenfassung

Allgemein wurden alle Testfälle bestanden, was darauf hindeutet, dass sich die Funktion "creatİn verschiedenen Szenarien wie erwartet verhält.

15 Testdurchführung (2023-07-12)

Art des Tests: Unit-Test

Ausgeführte Testfälle: ?? Beteiligte Tester: Qiyue Zhang

Abgedeckte Funktionen: **F2**

15.1 Testumgebung

Die Testumgebung ist Jest, ein beliebtes JavaScript-Testframework, das häufig zum Testen von Node.js-Anwendungen verwendet wird.

15.2 Testprotokoll

Testfall	30
Tester	Qiyue Zhang
Eingaben	Es existieren drei Unfalldateien im ./mockData/accidents Ordner. Sende einen HTTP POST-Request mit einem Unfall, der zeitlich und geographisch gleich ist wie einer davon.
Soll - Reaktion	Der Unfall soll mit den existierenden Unfällen gruppiert werden.
Ist – Reaktion	Die searchAndGroupAccidents Funktion wird mit dem richtigen Ordnerpfad aufgerufen. Die distance Funktion wird einmal aufgerufen. Der Unfall wird korrekt gruppiert.
Ergebnis	Der Test war erfolgreich.

15.3 Zusammenfassung

Der Unfall wird korrekt gruppiert. Die Funktion verhält sich wie erwartet und gibt die entsprechenden Erfolgsmeldungen aus.