

Inhaltsverzeichnisse Beispiele

1. Präambel	7	2. Theoretische Konzepte	14	1. Präambel	9
1.1. Kurzfassung	7	2.1. REST-Paradigma	14	1.1. Team	9
1.2. Abstract	7	2.2. NoSQL Datenmodellierung	16	1.2. Danksagung	9
1.3. Team	8	2.2.1. Entwicklungsgrundlage der NoSQL-Datenmodellierung	16	1.3. Gendererklärung	9
1.4. Danksagung	8	2.2.2. Grundlagen der NoSQL Datenmodellierung	16		
1.5. Gendererklärung	8	2.2.3. NoSQL im CAP-Theorem	16		
2. Einleitung	9	2.2.4. NoSQL Datenmodelle	16		
2.1. Ausgangslage	9	2.3. Template-Engine	17	2. Ausgangssituation und Zielsetzung	10
2.1.1. Spezifische Ausgangslage - Oliver Fahrnik	9	2.4. React.js im Überblick	18	2.1. Problematik	10
2.1.2. Spezifische Ausgangslage - Armin Siegmeth	9	2.5. Satelliten- und Schadstoffdaten	19	2.2. Aufgabenstellung	11
2.2. Forschungsfrage	9	2.5.1. Kommunikation mit Satelliten	19	2.2.1. Mobile Applikation	11
2.2.1. Spezifische Forschungsfrage - Oliver Fahrnik	10	2.5.2. Satellitenbasierte Luftschadstoffmessung	19	2.2.2. Webapplikation	11
2.2.2. Spezifische Forschungsfrage - Armin Siegmeth	10	2.5.3. Luftschadstoffmessung am Beispiel der Sentinel-5P Mission	20	2.2.3. Technologie-Stack	12
3. Theoretische Grundlagen	11	2.6. Sentinel API	21		
3.1. Datenkonzepte zur Strukturierung und Speicherung von Daten	11	2.6.1. OGC API	21		
3.1.1. Relationale Datenbank	11	3. Analyse und Diskussion des State of the Art	24		
3.1.2. Dokumentorientierte Datenbank	17	3.1. Darlegung der Ausgangssituation	24	3. Erhebung potenzieller Technologien und deren Gegenüberstellung	13
3.1.3. Graphdatenbank	22	3.1.1. Technische Problemstellung	24	3.1. Plattformen zur Erstellung der mobilen Applikation	13
3.1.4. Datenbankkonzepte für stark vernetzte Daten	27	3.1.2. Wissenschaftliche Problemstellung	24	3.2. Vergleich zwischen Authentifizierungssystemen	14
3.2. Visualisierung von Daten	29	3.2. Diskussion relevanter Lösungsansätze	26		
3.2.1. Richtlinien	29	3.2.1. Aggregation der Satellitendaten	26		
3.2.2. Gestaltpsychologie	30	3.2.2. Direkte Abfrage von der Sentinel-API durch den Endnutzer	26		
3.2.3. Visualisierungsformen	33	3.2.3. Analyse der Bilddaten	26	4. Theoretische Grundlagen	17
3.2.4. Visualisierungsformen von stark vernetzten Daten	42	3.2.4. Auswahl des präferierbaren Lösungsweges	26	4.1. Design Prinzipien	17
4. Dokumentation der Implementierung	45	3.2.5. Visualisierung des präferierbaren Lösungsweges	27	4.2. Chart.js	18
4.1. Dokumentation - Grundlegend	45	4. Definition der technischen Anforderungen	29	4.3. Bootstrap	19
4.1.1. Testumgebung	45	4.1. Funktionale Anforderungen	29	4.4. Laravel	19
4.1.2. Technologien	45	4.2. Nichtfunktionale Anforderungen	29	4.5. MVC-Konzept	23
4.2. Dokumentation - Funktionen	53	4.3. Designanforderungen	30	4.6. Authentifizierung	24
4.2.1. Benutzermanagement	53	5. Systemdesign	31	4.7. Plesk	26
4.2.2. Einfache Konfiguration	54	5.1. Darstellung der Systemarchitektur	31		
4.2.3. Künstliche Intelligenz	56	5.2. Datenmodelle	32	5. Design des Systems	28
4.2.4. Datenvisualisierung	57	5.3. Export API	37	5.1. Datenbank	28
4.2.5. Reports	59	5.3.1. Satelliten Daten	37	5.2. API	29
4.2.6. Templates	60	5.3.2. Pollution Patrol Abfragen	38	5.3. Design des User Interfaces	31
5. Zusammenfassung und Ausblick	62	5.3.3. Suchmasken	41	5.3.1. Styleguide	31
5.1. Zusammenfassung	62	5.3.4. Suchmaskenergebnisse	44	5.3.2. Mobile Applikation	32
5.2. Ausblick	62	5.4. Webanwendung	46	5.3.3. Webapplikation	38
I. Literaturverzeichnis	63	5.4.1. Mockups	46		
II. Abbildungsverzeichnis	66	5.4.2. Styleguide	51		
		6. Implementierung	52		
		6.1. Entwicklungswerkzeuge	52		
		6.1.1. Visual Studio Code	52		
		6.1.2. Visual Studio Code Extensions	53		
		6.1.3. MongoDB Compass	53		
		6.2. Server	56		
		6.2.1. Node.js	56		
		6.2.2. Einrichtung des Servers	56		
		6.2.3. Datenbankintegration	59		
		6.2.4. Basis-Klasse für API-Abfragen	71		
		6.2.5. Implementierung der Worker Threads	73		
		6.3. Webanwendung	80		
		6.3.1. Einrichtung der Express.js-Template-Engine	80		
		6.3.2. Einrichten eines React-Typescript Projektes	81		
		6.3.3. User Interface	82		
		6.4. Export API	93		
		6.4.1. Einrichtung des Express.js-Projektes	93		
		6.4.2. Schnittstellen	95		
		7. Anwender Tests	101		
		8. Installation und Inbetriebnahme des Software-Prototyps	102		
		8.1. Voraussetzungen	102		
		8.2. Konfigurieren der Datenbank	102		
		8.3. Starten des Programms	102		
		9. Zusammenfassung und Ausblick	103		
		9.1. Zusammenfassung der Arbeit	103		
		9.2. Ergebnisse im Überblick	103		
		9.3. Herausforderungen	103		
		9.3.1. Herausforderungen im Team	104		
		9.3.2. Technische Herausforderungen	104		
		10. Arbeitseinteilung	105		
		10.1. Arbeitseinteilung Verfasser der Kapitel	105		
		I. Literaturverzeichnis	107		
		II. Abbildungsverzeichnis	108		
				7. Resümee und Ausblick	95
				7.1. Zusammenfassung	95

Inhaltsverzeichnisse Beispiele 2

1. Präambel	10	7. Die Turing-Maschine	45
1.1. Kurzfassung	10	7.1. Relevanz der Turing-Maschine und Turing-Vollständigkeit	45
1.2. Abstract	10	7.2. Herkunft und Bedeutung	45
1.3. Team	10	7.3. Funktionsweise der Turing Maschine	45
1.4. Danksagung	10	7.4. Turing-Vollständigkeit	46
1.5. Gendererklärung	10	7.5. Bedeutung der Turing-Vollständigkeit für die Softwareentwicklung	46
2. Einleitung	11	7.6. Nicht Turing-vollständige Sprachen	46
2.1. Ausgangslage	11	8. Entwurf eines Interpreters	48
2.1.1. Umfeldeingrenzung	11	8.1. Grundlagen eines Interpreter	48
2.1.2. Spezifische Ausgangslage - Konstantin Kanzler	11	8.2. Entwurfsmuster zur Implementation eines Interpreters	48
2.1.3. Spezifische Ausgangslage - Alexander Dick	11	8.3. Funktionsweise des Entwurfsmusters	49
2.2. Forschungsfrage	12	8.4. Einsatzmöglichkeiten des Interpreters	50
2.2.1. Spezifische Forschungsfrage - Alexander Dick	12	8.5. Bestandteile eines Code-Interpreters	50
2.2.2. Spezifische Forschungsfrage - Konstantin Kanzler	12	8.5.1. Lexer	50
3. Konzepte hinter Programmiersprachen	13	8.5.2. Parser	50
3.1. Arten von Programmierparadigmen	13	8.5.3. AST	50
3.1.1. Imperative Programmierung	13	9. Implementation des Interpreters	51
3.1.2. Deklarative Programmierung	18	9.1. Ablauf des Interpret-Prozesses	51
3.2. Grundlegende Programmierkonzepte	20	9.2. Lexer	51
4. Evaluierung der Grundkonzepte im softwaretechnischen Spannungsfeld	30	9.3. Parser	59
4.1. Anweisungen	30	9.9. Parser	59
4.1.1. Anweisungsarten	30	9.10. Ablauf des Parser-Prozesses	60
4.2. Funktionen	30	9.11. Diagramm des Ablaufs	60
4.3. Variablen/Arrays	31	9.12. Implementationen von Parser-Statements	61
4.4. Bedingungen	31	9.12.1. Typen von Parse-Statements	62
4.5. Schleifen	31	10. Technologien des Backends	65
4.6. Evaluierungsergebnis	31	10.1. Einleitung und Motivation	65
5. Katalog der Stufenkonzepte	32	10.2. ASP.NET Core	65
5.1. Definition der Konzepte	32	10.3. Domain Driven Design	65
5.2. Funktionen	33	10.4. Struktur des Backend-Systems	66
5.3. Schleifen	34	10.5. Authentication innerhalb des Systems	66
5.4. Ausgaben	35	10.6. Datenbank	67
5.5. If-Bedingungen	36	11. Evaluierung Imperativer und Deklarativer Programmiermethodiken	68
5.6. Variablen	37	11.1. Anliegen und Motivation	68
5.7. Arrays	38	11.2. Imperative Programmierung	68
6. Implementation der Benutzeroberfläche	39	11.3. Deklarative Programmierung	68
6.1. Aufbau der Benutzeroberfläche	39	11.4. Schlussfolgerung	69
6.1.1. Funktionsweise der Simulationsfläche	40	11.5. Festlegung einer Programmierungsart für Programmieranfänger	69
6.1.2. Funktionsweise der Interaktionsschnittstelle	42	11.6. Beleg der Turing-Vollständigkeit der abstrakten Programmiersprache	70
6.2. Verwendete Technologie	43	12. Installation	73
		12.1. Mindestanforderungen	73
		12.2. Systemempfehlungen	73
		12.3. Starten der Applikation	73
		12.4. Einrichtung der Datenbank	73
		12.5. Benutzen der Applikation	73
		13. Zusammenfassung und Ausblick	74
		13.1. Aufgabe der Diplomarbeit	74
		13.2. Vorbereitung	74
		13.3. Umsetzung	74
		13.4. Auslieferung	74
		13.5. Ausbaupotential	74