

Entwicklung einer Analyse-Applikation inklusive statistischer Auswertung von Zugfahrzeiten innerhalb von Testszenarien

Dokumentation der betrieblichen Projektarbeit von Diana Tashkulova zum Erlangen des
Abschlusses als Fachinformatikerin Fachrichtung Anwendungsentwicklung

durchgeführt im IT-Ausbildungsverbund der
Technischen Universität Darmstadt
Karolinenplatz 5
64289 Darmstadt



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Mit der freundlichen Bitte um Benotung.

Privatanschrift Auszubildende
Diana Tashkulova
Pruthstraße 4
64295 Darmstadt
Tel.: +49 157 71002474
E-Mail: diana.tashkulova@tu-darmstadt.de

Ausbilder

Peter Knuth
Technische Administration
Otto-Berndt-Straße 2
64287 Darmstadt
Tel.: +49 6151 16-65922
E-Mail: peter.knuth@tu-darmstadt.de

Projektverantwortlicher

Peter Knuth
Technische Administration
Otto-Berndt-Straße 2
64287 Darmstadt
Tel.: +49 6151 16-65922
E-Mail: peter.knuth@tu-darmstadt.de

Persönliche Erklärung der Prüfungsteilnehmerin/des Prüfungsteilnehmers

Ich versichere durch meine Unterschrift, dass ich die betriebliche Projektarbeit und die dazugehörige Dokumentation selbstständig in der vorgesehenen Zeit erarbeitet habe.

Ebenso bestätige ich, dass ich bei der Erstellung der Dokumentation meiner Projektarbeit weder Teile noch vollständige Passagen aus Projektarbeiten übernommen habe, die bei der prüfenden oder einer anderen Kammer eingereicht wurden.

Darmstadt, 15.04.2021

Ort, Datum



Diana Tashkulova

Inhaltsverzeichnis

1.....Ausgangssituation	3
1.1 Abweichung vom Projektantrag	3
1.2 Projektziele	3
1.3 Projektbeschreibung aus Kundensicht	3
1.4 Projektumfeld	4
1.5 Stakeholderanalyse	4
1.6 Machbarkeitsstudie	6
1.7 Risikomindernde Maßnahmen	7
1.8 Prozessschnittstellen	8
2.....Ressourcen- und Ablaufplanung	8
2.1 Entscheidungsmatrix	8
2.2 Ablaufplan	9
2.3 Definition von Standards	10
2.4 Ressourcenplanung	10
2.4.1 Personalplanung	10
2.4.2 Sachmittelplanung	10
2.4.3 Kosten-Nutzenplanung	11
3.....Durchführung und Auftragsbearbeitung	13
3.1 Prozessschritte, Vorgehensweise und Qualitätssicherung	13
3.1.1 Projektplanung	13
3.1.2 Projektrealisierung	14
3.1.3 Test und Qualitätssicherung	18
3.1.4 Inbetriebnahme	18
3.1.5 Erstellen einer Projektdokumentation	18
3.2 Funktionstest	18
3.3 Abweichungen und Anpassungen	20
4.....Projektergebnisse	21
4.1 Soll-Ist-Vergleich	21
4.2 Übergabe des Projektergebnisses	21
4.3 Fazit zum Projekt	21
4.4 Ausblick	21
5.....Abbildungsverzeichnis	22
6.....Tabellenverzeichnis	22
7.....Literaturverzeichnis	22
8.....Abkürzungsverzeichnis	23
9.....Glossar	24
10....Anhang	i
10.1 Kundendokumentation	i
10.1.1 Statistische Werte laden	i
10.1.2 Statistische Werte löschen	i

10.1.3	Statistische Werte auswählen	i
10.1.4	Diagramme als PDF speichern	i
10.1.5	Anleitung lesen	i
10.2	Auszug aus dem Lastenheft	ii
10.3	Auszug aus dem Pflichtenheft	iii
10.4	UML-Diagramme	iv
10.4.1	Klassendiagramm	iv
10.5	Struktur der Logdatei	v
10.6	POM-Struktur	vi
10.7	Hibernate Konfiguration File	vii
10.8	Struktur	viii
10.9	Auszug aus dem Quellcode	ix

1 Ausgangssituation

1.1 Abweichung vom Projektantrag

Es liegen keine Abweichungen zum Projektantrag „Entwicklung einer Analyse-Applikation inklusive statistischer Auswertung von Zugfahrtzeiten innerhalb von Testszenarien“ vor.

1.2 Projektziele

In Rahmen des Forschungsprojekts smartLogic am Institut für Bahnsysteme und Bahntechnik, welches eine regelbasierte Sicherungslogik für neue Generation des plattformunabhängigen Stellwerks repräsentiert, soll eine Analyse-Applikation für die grafische Auswertung und sicherer Speicherung von statistischen Werten von Zugfahrtzeiten innerhalb von Testszenarien entwickelt werden.

Ziel des Projekts ist die Bereitstellung der grafischen Darstellungen von statistischen Ergebnissen, welche die Möglichkeit schafft, ein effizientes Szenario mit optimalen Zugfahrtzeiten unter bestimmten Fahrregeln schnell zu ermitteln. Auf Basis dieser Ergebnisse werden entsprechende Schlüsse hinsichtlich der getesteten Fahrregel gezogen.

Die Analyse-Applikation soll in der Programmiersprache Java implementiert und nach dem Model-View-Controller-Entwurfsmuster (MVC) realisiert werden. Die grafische Darstellung erfolgt mittels des JFreeChart Frameworks, welche passende Bibliotheken für verschiedene Diagrammtypen liefert. Um die statistischen Ergebnisse in einer SQL Datenbank zu speichern und sie bei Bedarf aus der Datenbank abzurufen, wird das Hibernate Framework für die Datenbankbindung verwendet.

Die notwendigen Informationen und die zur Auswertung benötigten Testdaten werden von wissenschaftlichen Mitarbeitenden und studentischen Hilfskräften zur Verfügung gestellt.

Das Projekt wird von der Auszubildenden, Frau Diana Tashkulova, durchgeführt.

1.3 Projektbeschreibung aus Kundensicht

Aktuell wird die Analyse der Zugfahrtzeiten aufgrund des Fehlens von passenden Werkzeugen manuell durchgeführt. Die statistischen Werte werden von der Fahrdynamik der Modellanlage im Eisenbahnbetriebsfeld erstellt und in Logdateien zusammengefasst, die dann entsprechend von wissenschaftlichen Mitarbeitenden (WiMis) und studentischen Hilfskräften (HiWis) vor Ort ausgewertet werden.

Für die manuelle Auswertung sollen die Datensätze von Testszenarien pro Simulation gesammelt, zusammengerechnet, ausgewertet und zum Schluss in einer Excel-Tabelle gespeichert werden. Da die Logdateien i.d.R. aus größeren und komplexen Datensätzen bestehen, führt es im Fall der Bearbeitung von statistischen Datensätzen zu einer dauerhaften Analyse, während deren Ausführung mehrere manuelle Datenauswertungsfehler oft nicht erkannt bzw. nicht behoben werden.

Um die Analyse- und Auswertungsprozesse zukünftig zeiteffizient und einwandfrei durchführen zu können, sollen sie durch eine programmierte Analyse-Applikation erfolgen, welche zusätzlich eine grafische Aufbereitung von statistischen Werten bereitstellt. Die Forschungsbeteiligten können daher von der aufwändigen Untersuchung befreit werden und stattdessen sich auf wichtige Forschungsprozesse konzentrieren.

Aus Kundensicht gilt das Projekt als erfolgreich abgeschlossen, wenn alle Kundenanforderungen erfüllt und die Übergaben unter Beachtung von Abnahmekriterien erfolgt wurden.

1.4 Projektumfeld

Die Technische Universität Darmstadt (TU Darmstadt) bildet mit der Goethe-Universität Frankfurt am Main und der Johannes-Gutenberg-Universität Mainz eine Gemeinschaft der Rhein-Main-Universitäten. Ihr Mittelpunkt liegt hauptsächlich auf den technischen Disziplinen, bietet aber auch natur-, geistes-, und humanwissenschaftliche Studienfächer an. Die TU Darmstadt besteht aus 13 fach- und fünf ingenieurwissenschaftlich geprägten Studienbereiche, die zusammen 113 Studiengänge umfassen. Zurzeit hat die TU Darmstadt 25.170 Studierende und 4.991 Mitarbeitende.

Das Hochschulrechenzentrum (HRZ) ist das Kompetenzzentrum der TU Darmstadt in Bereich der Informationstechnologie. Das HRZ stellt der TU Darmstadt die gesamten IT-Dienste im Rahmen der Forschung, Lehre und des Studiums zur Verfügung.

Das Institut für Bahnsysteme und Bahntechnik (BST) bildet zusammen mit zwei weiteren Instituten und einer Fachgruppe einen Institutsverbund für Verkehr. Am BST werden Forschungen zu innovativen Entwicklungen und Ermittlungen von Lösungen bei Störfällen im Schienenverkehr durchgeführt. Zusammen mit Partner-Unternehmen DB Training, Learning & Consulting und Akademischer Arbeitskreis Schienenverkehr e.V. (Aka Bahn) führt das BST das Eisenbahnbetriebsfeld Darmstadt (EBD).

Das EBD ist eine gemeinsam entwickelte Modelbahnanlage, in der neue Betriebsverfahren mithilfe von verschiedenen Stellwerken erprobt, untersucht sowie weiterentwickelt werden. In der Lehre wird das EBD für Demonstration von Eisenbahnbetriebsprozessen sowohl für Studierende, als auch für externe Gäste während Lernveranstaltungen im Bereich Eisenbahnsicherungstechnik genutzt.

1.5 Stakeholderanalyse

Um die relevanten Interessengruppen und deren Auswirkung im Projektprozess festzustellen, wird eine Stakeholderanalyse durchgeführt. Die Auswirkungsstufe der Projektbetroffenen und -beteiligten ergibt sich aus ihrem Haltungs- und dem Einflussgrad bezüglich des Projekts. Dadurch wird festgestellt, welche Personengruppen eine positive, negative sowie neutrale Auswirkungen haben und können dementsprechend behandelt werden.

Stakeholderanalyse		
Personenkreis	Haltung	Einfluss
Professor des BST	Positiv (5)	Sehr wenig (1)
WiMi	Positiv (5)	Sehr hoch (5)
HiWi	Leicht positiv (4)	Mittlerer (3)
Technische Mitarbeitende	Leicht positiv (4)	Wenig (2)
Legende: Die Haltung ist in die Stufen „Negativ“ (1), „Leicht negativ“ (2), „Neutral“ (3), „Leicht positiv“ (4), „Positiv“ (5) eingestellt. Der Einfluss ist in Stufen „Kein“ (0), „Sehr wenig“ (1), „Wenig“ (2), „Mittlerer“ (3), „Hoher“ (4), „Sehr hoch“ (5) eingestellt.		

Tabelle 1: Stakeholderanalyse. (Quelle: Eigene Darstellung)

Großes Interesse an die Projektrealisierung haben wissenschaftliche Mitarbeitende, die das Projekt beauftragen haben. Dabei soll die Haltung des Professors des BST in Bezug der Projektumsetzung berücksichtigt werden, da er die Interessen des Institutes von Bahnsysteme und Bahntechnik vertritt, in dessen Forschungsgebiet das Projekt umgesetzt wird.

Es ist ebenso wichtig, dass technische Mitarbeitende ins Projekt einbezogen werden, da die Modellanlage, auf der die Simulationen stattfinden, und der Server im EDB, auf dem die Datenbank des Projekts zukünftig eingesetzt wird, von ihnen technisch betreut und betrieben werden.

Die studentischen Hilfskräfte haben ebenfalls eine positive Haltung bezüglich des Projektprodukts, da sie bei der Auswertung von komplexen Datensätzen zukünftig unterstützt werden.

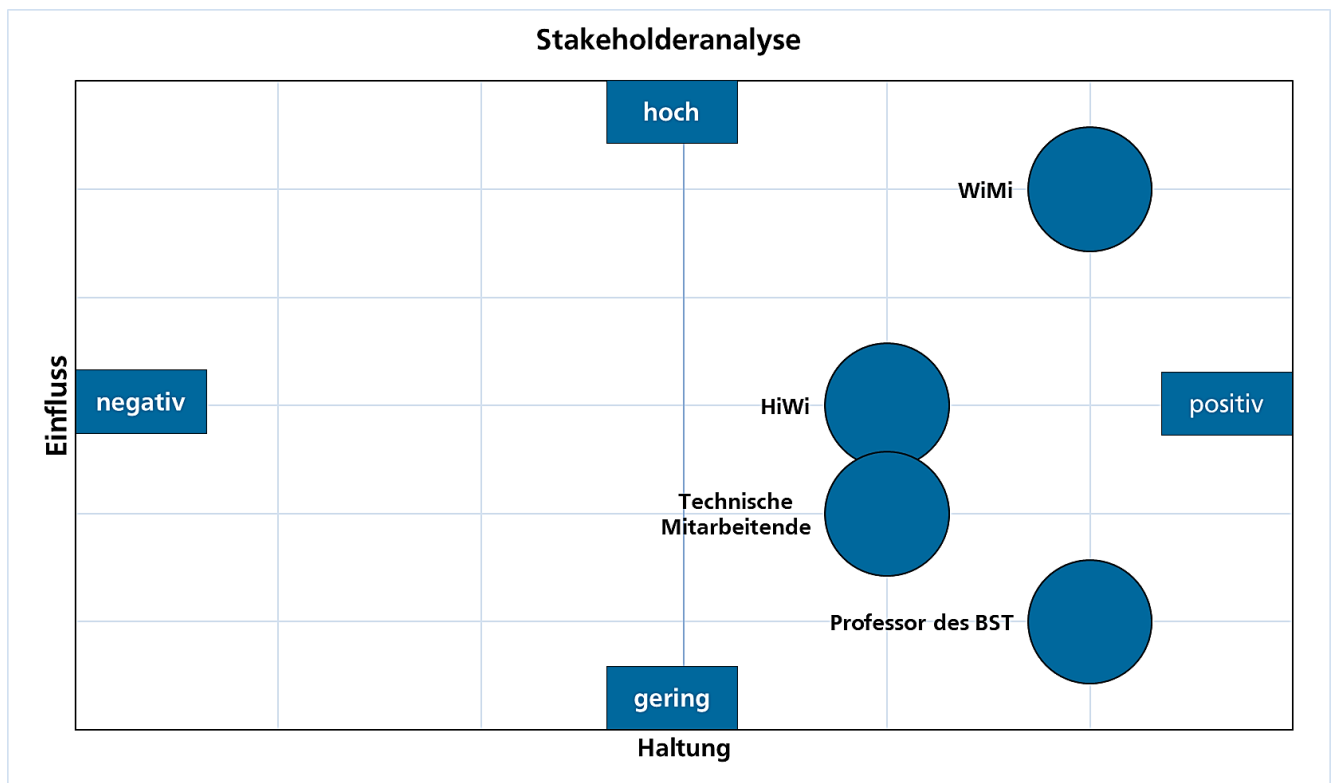


Abbildung 1: Stakeholdermatrix. (Quelle: Eigene Darstellung)

Ausgehend von den Ergebnissen aus der Stakeholderanalyse werden die entsprechenden Maßnahmen bezüglich der Stakeholder unternommen:

- Die Stakeholder, die ein großes Interesse und ebenso einen großen Einfluss haben, werden am meisten bezüglich der Projektprozesse einbezogen.
- Die Stakeholder, die ein großes Interesse aber wenig Einfluss auf die Projektumsetzung haben, werden nur über die Projektschritte informiert.

1.6 Machbarkeitsstudie

Um die zur Umsetzung benötigten Ressourcen nach Verfügbarkeit zu prüfen und daraus auftretenden Risiken frühzeitig zu erkennen, soll eine Machbarkeitsstudie durchgeführt werden. Dadurch werden alle relevanten Projektaspekte untersucht, die den Projekterfolg möglicherweise verhindern können.

Wirtschaftliche Machbarkeit

Die zur Projektdurchführung anfallenden Kosten werden vom Institut für Bahnsysteme und Bahntechnik übernommen. Dazu sind die Personalkosten zu zählen, in denen die administrativen und Gemeinkosten pauschal mitgerechnet werden. Die Beschaffungskosten für Hard- und Software entfallen, da die zur Projektumsetzung benötigten Mittel bereits im Betrieb vorhanden sind oder eine Open-Source-Lizenz besitzen. Es werden daher keine zusätzlichen Kosten von dieser Seite her entstehen.

Technische Machbarkeit

Laut der Kundenanforderungen soll die Implementierung der Analyse-Applikation in aktueller Version der Java Programmiersprache mit einer MySQL Datenbankanzbindung erfolgen. Für die Herstellung einer Datenbankverbindung sowie für die Ausführung der CRUD-Operationen wird das Hibernate Framework verwendet. Die grafische Auswertung von statistischen Datensätzen soll mittels von JFreeChart Drittanbieterbibliotheken entwickelt werden.

Dabei ist darauf zu achten, dass die Technologien im Zusammenhang keine Kompatibilitätsprobleme erzeugen und somit keine extra Anpassungen benötigen.

Organisatorische Machbarkeit

Es bestehen ausreichend personelle Ressourcen, die bei Fragen zur Verfügung stehen. Des Weiteren wurden Vertretende des Auftraggebers im Fall von Nichterreichbarkeit aufgrund von Urlaub, Krankheit oder andere Geschehen festgelegt.

Damit soll sichergestellt werden, dass keine Gefährdungen durch organisatorische Mängel während der Projektdurchführung eintreten.

Juristische Machbarkeit

Es werden keine personenbezogenen Daten durch die Analyse-Applikation verarbeitet oder gespeichert. Daher ist keine Abstimmung mit den Datenschutzbeauftragten der TU Darmstadt notwendig.

Risikoanalyse

Die Risikoanalyse dient dazu, eventuelle Risiken, die eine Auswirkung auf die Projektumsetzung haben, zu untersuchen und bei Bedarf zu beseitigen.

In folgender Tabelle werden alle möglichen Risiken nach Eintrittswahrscheinlichkeit und erwarteten Schadenshöhen analysiert und bewertet. Bei größeren Schadensereignissen sind die passenden Handlungsbedarfe zu deren Senkung bzw. Verhinderung erforderlich.

Risikoanalyse			
Risikotyp	Wahrscheinlichkeit	Auswirkung	Bewertung
Ressourcen sind nicht ausreichend	Niedrig (2)	Hoch (4)	Geringerer Handlungsbedarf (8)
Dauerhafte Internetverbindung ist unzuverlässig	Sehr niedrig(1)	Mittel (3)	Kein Handlungsbedarf (3)
WiMis sind für Rückfragen nicht erreichbar	Sehr niedrig (1)	Hoch (4)	Kein Handlungsbedarf (4)

Datenformat für Auswertungen wurde geändert	Niedrig (2)	Hoch (4)	Geringerer Handlungsbedarf (8)
Daten werden inkorrekt bearbeitet	Sehr niedrig (1)	Hoch (4)	Kein Handlungsbedarf (4)
Ergebnisse sind grafisch falsch dargestellt	Sehr niedrig (1)	Hoch (4)	Kein Handlungsbedarf (4)
Datenbankverbindung kann nicht hergestellt werden	Niedrig (2)	Hoch (4)	Geringerer Handlungsbedarf (8)
Daten werden nicht in die Datenbank gespeichert und aus der Datenbank abgerufen	Sehr niedrig (1)	Hoch (4)	Kein Handlungsbedarf (4)
Deadlines werden nicht eingehalten	Mittel (3)	Niedrig (2)	Geringerer Handlungsbedarf (6)
Legende: Die Wahrscheinlichkeit und Auswirkung sind in Stufen 1 (Sehr niedrig), 2 (Niedrig), 3 (Mittel), 4 (Hoch), 5 (Kritisch) eingestellt. Die Bewertung ist in Stufen 1-5 (Kein Handlungsbedarf), 6-10 (Geringerer Handlungsbedarf), 11-15 (Handlungsbedarf), 16-20 (Hoher Handlungsbedarf), 20-25 (Kritischer Handlungsbedarf)			

Tabelle 2: Risikoanalyse. (Quelle: Eigene Darstellung)

1.7 Risikomindernde Maßnahmen

Um die potenziellen Risiken frühzeitig zu verhindern oder gezielt zu beherrschen, sollen bestimmte Maßnahmen unternommen werden:

- Um zu prüfen, ob es ausreichende Ressourcen zur Verfügung stehen, wird eine Machbarkeitsanalyse durchgeführt. Diese soll sicherstellen, dass es keine Mängel an benötigten Ressourcen bestehen und das Projekt somit reibungslos starten kann.
- Es wird ein Zeitraum von fünf Wochen für die Umsetzung festgelegt, um eine Pufferzeit aufzubauen, falls die Projektverantwortlichen oder Ansprechpartner aufgrund von Urlaub, Krankheit oder anderen Terminen abwesend sind. Dies schafft eine Flexibilität in der Projektplanung, wobei die Projektvorgänge ohne Einfluss auf das Gesamtprojekt zeitlich verschoben und erledigt werden können. Daneben hilft es passende Termine für Rücksprache anzulegen, damit alle Projektbeteiligte teilnehmen können.
- Sowohl programmierte Schnittstellen, als auch die Anwendungsfälle der Analyse-Applikation werden mittels Funktionstests mehrfach geprüft, um die Ursachen des Fehlereintritts zu eliminieren und letztlich auszuschließen.
Die Funktionalitäten der Applikation werden anhand der vorgesehenen Testszenarien getestet und die daraus entstandenen Ergebnisse anschließend in einem Testprotokoll zusammengefasst.
- Es soll genug freie Speicherkapazitäten auf dem Server des EBD zur Verfügung stehen, damit die statistischen Ergebnisse in die Datenbank eingespielt werden können. Dies wird von technischen Mitarbeitenden konfiguriert und zur Nutzung freigegeben.

1.8 Prozessschnittstellen

Rollenzuweisung	
Aufgabenfeld	Ansprechpartner
Auftraggeber	Institut für Bahnsysteme und Bahntechnik TU Darmstadt
Vertretende des Auftraggebers	M. Sc. Frederik Döpmeier Dr. rer. nat. Georg F.Bolz
Projektbetreuer	Peter Knuth
Projektmitarbeiterin	Diana Tashkulova

Tabelle 3: Rollendarstellung aller Personen, die an der Projektdurchführung beteiligt sind. (Quelle: Eigene Darstellung)

Das Projekt beginnt mit der Analyse der aktuellen Auswertungsprozesse der statistischen Werte, die für die Feststellung von möglichen Anwendungsfällen der Applikation zugrunde liegen. Das Projekt wird von Projektmitarbeiterin Diana Tashkulova in Rücksprachen mit anderen Projektbeteiligten geplant und selbständig durchgeführt.

Der wissenschaftliche Mitarbeiter M. Sc. Frederik Döpmeier vertritt das Interesse des BST und legt daher die Anforderungen bezüglich der Funktionalität der Applikation und die Übergabekriterien des Projektergebnisses fest. Herr Döpmeier besitzt die für Projektdurchführung benötigte Qualifikation und Kenntnisse und steht bei Fragen als Ansprechpartner zur Verfügung.

Dr. Bolz stellt die passenden Testdaten bereit und unterstützt bei Fragen bezüglich der Datenauswertung. Bei Bedarf übernimmt Dr. Bolz den Posten des WiMis.

Herr Peter Knuth steht zur Unterstützung und zur Koordinierung während der Projektumsetzung. Als IT-Administrator betreibt Herr Knuth den Server im EBD, auf dem die Datenbank für das Speichern der statistischen Ergebnisse eingesetzt wird.

2 Ressourcen- und Ablaufplanung

2.1 Entscheidungsmatrix

Eine Entscheidungsmatrix bezüglich der zur Projektimplementierung eingesetzten Technologien konnte von der Projektmitarbeiterin nicht angefertigt werden, da diese vom Kunden bereits festgelegt wurden. Da die meisten im Eisenbahnbetriebsfeld laufenden Applikationen und Programme Java basierend sind, wurde hier vorgegeben die Java-Technologie bzw. Java Development Kit (JDK) in aktuellsten Version zu verwenden. Die grafische Auswertung laut der Kundenanforderungen soll mittels JFreeChart Bibliotheken erfolgen. Um einen klaren und einfachen Umgang mit der MySQL-Datenbank zu gewährleisten, wird das Hibernate Framework in dem Projekt eingesetzt. Dies ermöglicht, die Datenbankverbindung reibungslos herzustellen und die Vorgänge zum Speichern, Löschen, Abrufen von Datensätzen aus der Datenbank auszuführen.

Für eine reibungslose Projektumsetzung wird ein Zeitraum von fünf Wochen festgestellt, damit die Projektvorgänge in Fall von ungeplanten Umständen innerhalb von definierten Pufferzeiten ohne negative Auswirkung auf das Gesamtprojekt erledigt werden können.

2.2 Ablaufplan

Der Projektablauf wurde mittels eines Gantt-Diagramms gesteuert, welches die Projektphasen sowie deren Teilaufgaben zeitbezogen darstellt. Dies schafft einen schnellen Überblick über allen Aktivitäten des Projekts sowie über den aktuellen Stand der Projektdurchführung.

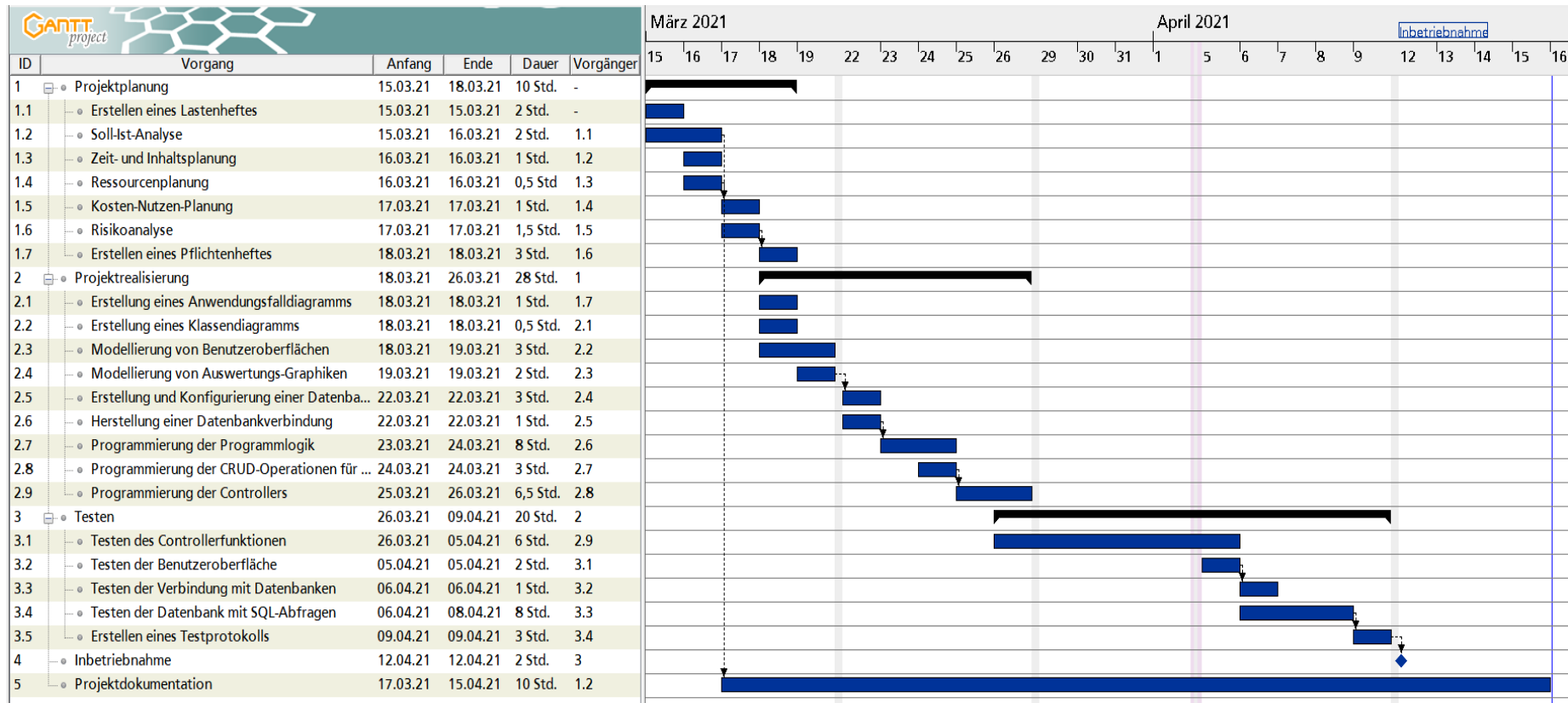


Abbildung 2: Zeitlicher Projekt-Ablaufplan. (Quelle: Eigene Darstellung. Erstellt mittels des Programms GanttProject 2.8.10 Pilsen unter General Public-Licence)

2.3 Definition von Standards

Wie bereits in Kapitel 1.6 erwähnt, werden keine personenbezogenen Daten durch die Analyse-Applikation verarbeitet. Daher ist nicht notwendig, eine Abstimmung mit den TU Datenschutzbeauftragten durchzuführen.

Die zur Projektumsetzung verwendeten Technologien besitzen eine Open-Source oder freie Software Lizenz und können unter der Beachtung der Open-Source Grundanforderungen vergütungsfrei verwendet werden. Eine komplette Auflistung von der in dem Projekt eingesetzten Software ist in Kapitel 2.4 unter Punkt Sachmittelplanung zu entnehmen.

2.4 Ressourcenplanung

2.4.1 Personalplanung

In der folgenden Tabelle sind die Stundenaufwendungen von Personen, die von der Projektdurchführung direkt betroffen wurden, zu sehen. Dies hilft, die gesamten Personalkosten zu ermitteln, die für die Projektumsetzung benötigt wurden.

Rollenzuweisung				
Personen-anzahl	Posten	Einzelkosten	Stundenanzahl	Gesamtkosten
1	Zeitstunden Auszubildende	25,00€	70 Stunden	1750,00€
1	Zeitstunden für Gespräche mit Projektbetreuer	36,16€	4 Stunden	144,64€
1	Zeitstunden für Gespräche mit Projektauftraggeber	40,73€	2 Stunden	81,46€
1	Zeitstunden für Gespräche mit Hilfwissenschaftler	11,75€	7 Stunden	82,25€
	Gesamtkosten			2.058,35€

Tabelle 4: Personalkostenplanung (Quelle: Eigene Darstellung)

2.4.2 Sachmittelplanung

Die Projektdurchführung wird mittels bereits vorhandener Ressourcen erfolgen.

Die anfallenden Kosten für die Nutzung von Büroräumen und -materialen sowie Gemeinkosten werden in der Kostenkalkulation des Projekts nicht berücksichtigt, da diese bereits in Personalkosten pauschal mitgerechnet werden.

Die verwendete Software und Drittanbieterbibliotheken werden unter den Open-Source-Lizenzen zur Verfügung gestellt und werden daher kostenfrei verwendet.

Für die Umsetzung wurden folgende Hardware, Software und Drittanbieterbibliotheken verwendet:

- Notebook Lenovo L460
- Windows 10 Betriebssystem
- Microsoft Office Professional Plus 2016
- Eclipse Entwicklungsumgebung Version 4.11.0
- Java Development Kit (JDK) Version 15.0.1
- XAMPP Paket Version 7.5.1
- JFreeChart Bibliotheken Version 1.0.13
- Hibernate Framework Version 5.4.27

- Apache Maven Build-Management-Tool Version 3.6.0
- GanttProject 2.8.10 Pilsen

Folgende Software wird unter Open-Source Lizenzen zur Verfügung gestellt, welche das Recht gewähren, die Software zu nutzen, zu modifizieren sowie weiterentwickeln.

- Eclipse IDE - Eclipse Public License.
- Java Development Kit - GNU General Public License.
- XAMPP - GNU General Public License.
- JFreeChart – GNU Lesser General Public License (LGPL).
- Hibernate - GNU Lesser General Public License (LGPL).
- Apache Maven - GNU General Public License.
- GanttProject 2.8.10 Pilsen - General Public License (GPL) Version 3.

2.4.3 Kosten-Nutzenplanung

Es ist grundsätzlich schwer, den durch die Nutzung der Analyse-Applikation entstehenden Gewinn abzuschätzen, da er von vielen beweglichen Faktoren abhängt. Dazu gehören die im EBD geplanten Termine für die Simulationen von Zugfahrten pro Jahr, Anzahl der testierenden Regelwerke pro Simulation und die Anzahl der verfügbaren WiMis und HiWis, die die Auswertung durchführen sollen. Da Termine für Zugfahrtsimulationen dementsprechend unterschiedlich ablaufen können, wurde die Kosten-Nutzenanalyse auf Basis von einer in Rücksprache mit WiMis und HiWis abgestimmten Abschätzung durchgeführt. Laut dieser Abschätzungsanalyse haben sich folgende Zahlenwerte gebildet:

Es werden zehn Termine für die Zugfahrtsimulationen jährlich stattfinden. Die davon erzeugten Logdateien mit statistischen Werten werden von einem eingestellten HiWi analysiert und ausgewertet. Dabei sollen ein WiMi und ein technischer Mitarbeiter im Prozess mitwirken und bei Bedarf unterstützen.

Aktuell werden für die manuelle Analyse, sowie den Auswertungsprozess pro Simulation jeweils ca. fünf Stunden des HiWis, 30 Minuten des WiMis und 30 Minuten des technischen Mitarbeiters benötigt.

Diese Prozesslaufzeit soll durch den Einsatz der Analyse-Applikation optimiert werden, indem die Auswertungen von statistischen Werten in kürzester Zeit einwandfrei erledigt werden. Außerdem wird keine Unterstützung durch den HiWi benötigt, da die zur Auswertung führenden Operationen von der Applikation allein gesteuert werden.

Stundenkosten pro Simulation im Vergleich			
Personen	Kosten pro Stunde	Aktueller Zeitausspruch in Stunden	Geplanter Zeitausspruch in Stunden
WiMi	54,30€	0,5 Stunden	0,5 Stunden€
Technische Mitarbeitende	36,16€	0,5 Stunden	0,5 Stunden€
HiWi	11,75€	5,5 Stunden	0 Stunden€
Gesamtkosten		109,86€	45,23€

Tabelle 5: Stundenkosten pro Simulation im Vergleich. (Quelle: Eigene Darstellung)

Bei der Kosten-Nutzenanalyse werden die Beschaffungskosten der Applikation im Wert von 2.058,35€ in den gezielten Gesamtkosten mitgerechnet. Daher wird die Kostenkalkulation im ersten Jahr des Applikationsbetriebs nach der folgenden Formel durchgeführt:

Beschaffungskosten + (Stundenkosten pro Simulation x Anzahl der Simulationen pro Jahr)

Kosten-Nutzen-Analyse						
Anzahl der Jahre	Anzahl der Simulationen pro Jahr	Gesamtkosten pro n Simulationen (Aktuell)	Gesamtkosten pro n Simulationen (Geplant)	Gesamtkosten nach n Jahren (Aktuell)	Gesamtkosten nach n Jahren (Geplant)	Gewinn & Verlust
1	10	1.098,60€	452,30€	1.098,60€	2.510,65€	-1.412,05€
2	10	1.098,60€	452,30€	2.197,20€	2.962,95€	-765,75€
3	10	1.098,60€	452,30€	3.295,80€	3.415,25€	-119,45€
4	10	1.098,60€	452,30€	4.394,40€	3.867,55€	+526,85€
5	10	1.098,60€	452,30€	5.493,00€	4.319,85€	+1.173,15€
6	10	1.098,60€	452,30€	6.591,60€	4.772,15€	+1.819,45€
7	10	1.098,60€	452,30€	7.690,20€	5.224,45€	+2.465,75€
8	10	1.098,60€	452,30€	8.788,80€	5.676,75€	+3.112,05€
9	10	1.098,60€	452,30€	9.887,40€	6.129,05€	+3.758,35€
10	10	1.098,60€	452,30€	10.986,00€	6.581,35€	+4.404,65€

Tabelle 6: Kosten-Nutzen-Analyse. (Quelle: Eigene Darstellung)

Anhand der Kosten-Nutzen-Analyse ist schnell zu ermitteln, dass die Gesamtkosten pro Jahr durch den Einsatz der Analyse-Applikation deutlich um mehr als Hälfte sinken. Dadurch werden die Kosten von ca. 646€ jährlich erspart und nach ca. vier Jahren ein Gewinn erwirtschaftet.

Zudem werden weitere nicht monetäre Vorteile geschaffen:

- die Vereinfachung und Beschleunigung des Auswertungsprozesses.
- die Bereitstellung von qualitativ höheren Ergebnissen.

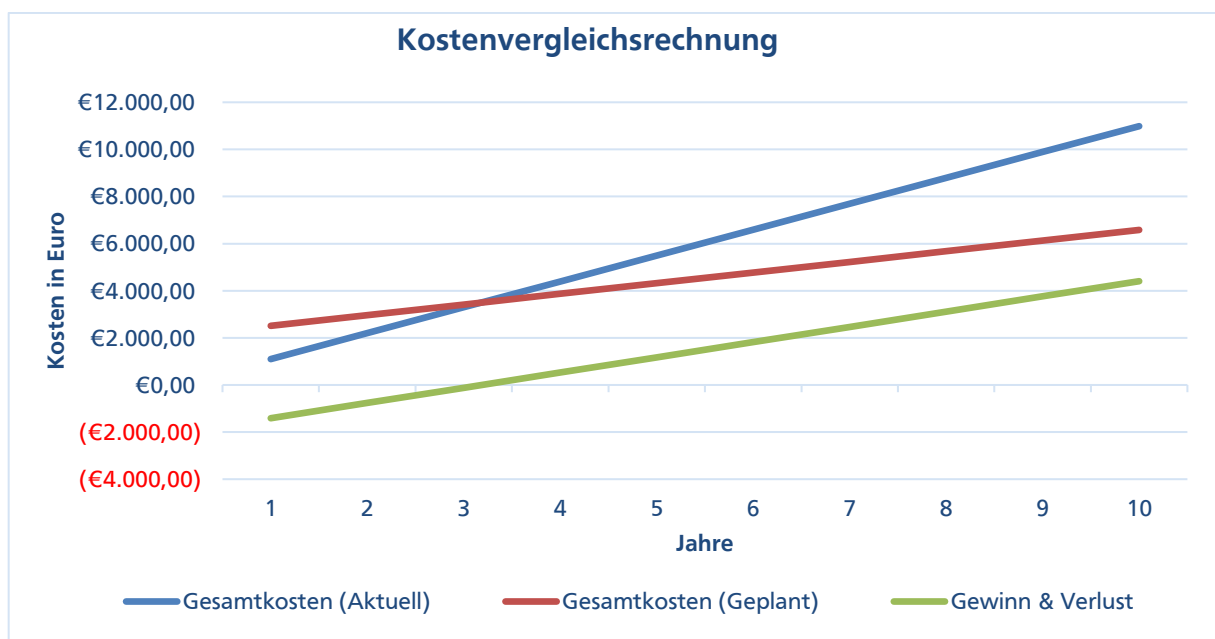


Abbildung 3: Kostenvergleichsrechnung. (Quelle: Eigene Darstellung)

3 Durchführung und Auftragsbearbeitung

3.1 Prozessschritte, Vorgehensweise und Qualitätssicherung

Es wurde ein klassisches Vorgehensmodell, das Wasserfallmodell, für die Projektentwicklung verwendet. Dieses verfügt unter anderem eine klare Abgrenzung von Projektphasen und ist einfach zu planen. Nach jeder Phase sollen vordefinierte Zwischenergebnisse (Meilensteine) erreicht werden, damit das Projekt reibungslos weiterentwickelt werden kann.

Die Entwicklungsprozesse wurden daher in fünf Phasen aufgeteilt.:

3.1.1 Projektplanung

- Soll-Ist-Analyse
- Kosten-Nutzen-Analyse
- Erstellen eines Pflichtenheftes

3.1.2 Projektrealisierung

3.1.2.1 Projektentwurf

- Festlegung einer Architektur der Analyse-Applikation
- Modellierung einer Benutzeroberfläche und der Grafiken
- Erstellung der UML-Diagramme
- Erstellung eines Entity-Relationship-(ERM) und eines relationalen Datenbankmodells (RM)
- Erstellung der Pseudocodes

3.1.2.2 Projektimplementierung

- Implementierung nach MVC Entwurfsmuster
- Erstellung einer MySQL-Datenbank
- Herstellung einer Verbindung mit der Datenbank
- Programmierung der Datenmodelle
- Programmierung der CRUD-Operationen
- Implementierung der Benutzeroberfläche und Grafiken
- Programmierung der Programmlogik

3.1.3 Test und Qualitätssicherung

- Durchführung eines Funktionstests
- Qualitätssicherung
- Erstellen eines Protokolltests

3.1.4 Inbetriebnahme

3.1.5 Erstellen einer Projektdokumentation

3.1.1 Projektplanung

Soll-Ist-Analyse

Anhand einer Soll-Ist-Analyse werden zu den in der Ist-Analyse entdeckten Schwachstellen und Problemen bei dem aktuellen Analysenprozess von statistischen Werten erwartete Lösungsansätze im Soll-Konzept entwickelt.

Kosten-Nutzen-Analyse

Hier sollen alle anfallenden Kosten, die zur Projektumsetzung benötigt wurden, berechnet und der Gewinn, welcher durch den Einsatz der Applikation entsteht, ermittelt werden.

Erstellen eines Pflichtenheftes

Im Pflichtenheft werden umsetzungsfähige Lösungsansätze für die Anforderungen, die im Lastenheft vom Kunden vorgegeben wurden, beschrieben. Mit der Akzeptanz des Pflichtenheftes durch den Auftraggeber beginnt die Umsetzung des Projekts.

3.1.2 Projektrealisierung

Die Projektrealisierung der Applikation erfolgt in zwei Phasen, in den Projektentwurfs- und Projektimplementierungsphasen. Einige während der Projektumsetzung entwickelten Entwurfsmuster, Konfigurationsdateien sowie Quellcodedokumentation wurden aufgrund der Übersichtlichkeit aus diesem Kapitel entnommen und sind daher im Anhang zu finden.

3.1.2.1 Projektentwurf

Auf Basis der im Pflichtenheft beschriebenen Anforderungen wird die Architektur der zu entwickelnden Applikation entworfen. Diese wird mittels Werkzeugen wie UML-Diagramme (Abbildung 5, Abbildung 9), ERM- und RM-Datenbankmodelle (Abbildung 6), Pseudocodes modelliert und anschließend mit einem Entwurf des grafischen Benutzerinterfaces ergänzt.

Festlegung der Architektur der Analyse-Applikation

Um die Architektur der Analyse-Applikation festzulegen, werden erst die grundlegenden Komponenten und deren Interaktion innerhalb des Systems erfasst. Dies hilft, um einen Überblick über die Funktionsweise der Applikation zu schaffen.

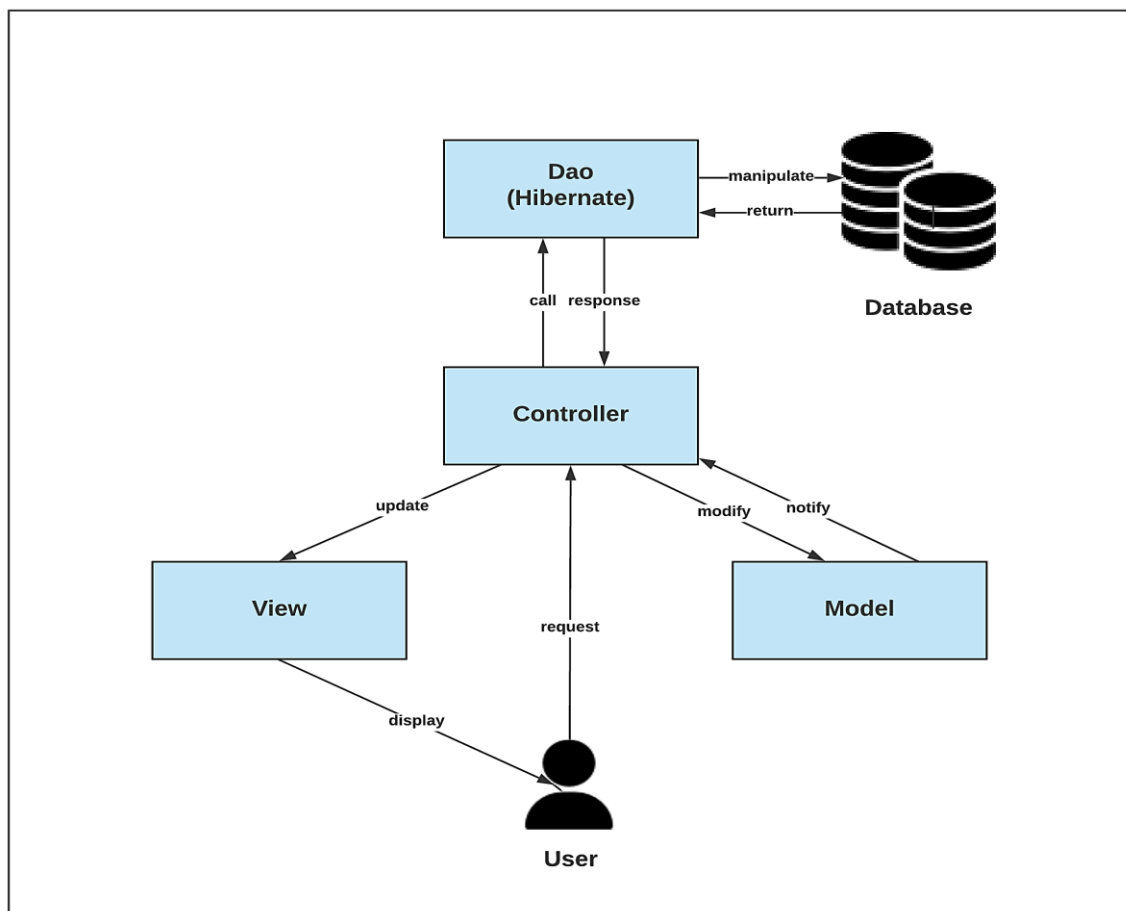


Abbildung 4: Architektur der Analyse-Applikation. (Quelle: Eigene Darstellung)

Modellierung der Benutzeroberfläche und der Grafiken

Es wird eine grafische Benutzeroberfläche modelliert, die zur Erleichterung der Nutzerbedienung der Applikation dienen soll. Zu diesem Zweck wird eine grafische Kontrollview zusammen mit zu verwendenden Diagrammen entworfen.

Erstellung der UML-Diagramme

Mithilfe der UML-Diagramme werden die möglichen Anwendungsfälle deutlicher analysiert und festgestellt. Es wurden mit einem, Anwendungsfall- und einem Klassendiagramm (Abbildung 9) zwei UML-Diagramme erstellt, mittels derer die Eigenschaften und Verhalten der Applikation erfasst und beschrieben werden. Diese werden als Grundlage für die Implementierung der Applikation verwendet.

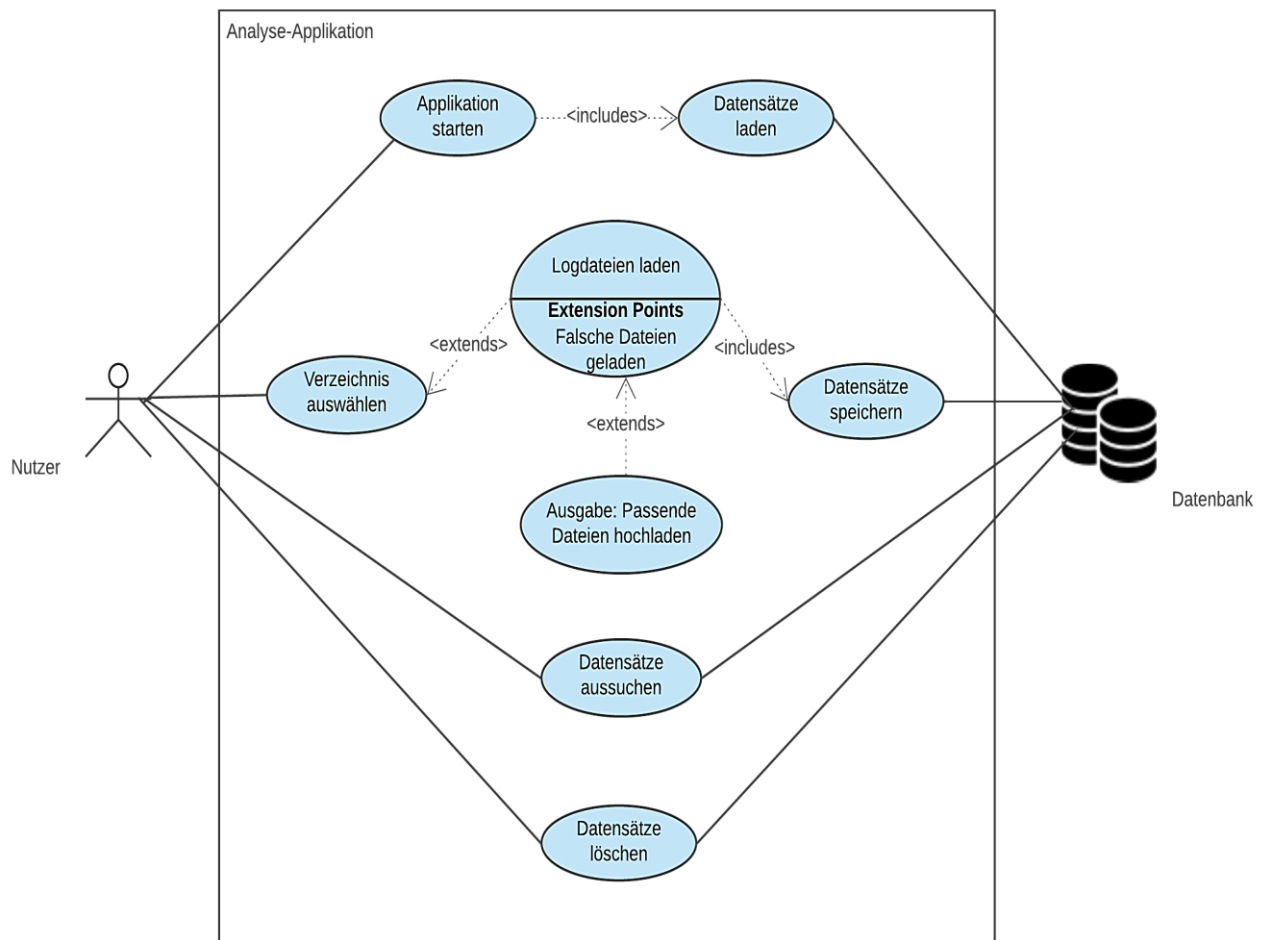


Abbildung 5: Anwendungsfalldiagramm der Analyse-Applikation. (Quelle: Eigene Darstellung)

Erstellung eines Entity-Relationship- und eines relationalen Datenbankmodells

Um eine leistungsfähige Datenbank anzulegen, wird diese mittels der Entity-Relationship- (ERM) und relationalen Datenbankmodelle (RM) untersucht und festgelegt.

Das Entity-Relationship-Modell ist erforderlich, um die Entitäten der Applikation und deren Beziehungen miteinander festzustellen.

Auf Basis dieses ERM wird ein relationales Datenbankmodell entworfen, welche eine detaillierte Beschreibung der zu entwickelnden Datenbank mit Tabellennamen, Attributen, Typen und Beziehungsarten darstellt.



Abbildung 6: Datenbankmodell der Analyse-Applikation. (Quelle: Eigene Darstellung)

Erstellung der Pseudocodes

Für bereits im Klassendiagramm definierten Methoden der einzelnen Klassen werden passende Algorithmen mittels Pseudocodes untersucht und entwickelt. Dies hilft schrittweise nachvollziehen, wie die Methoden ablaufen und wie die Daten bearbeitet werden sollen.

3.1.2.2 Projektimplementierung

In der Implementierungsphase (Anhang 10.8, Abbildung 11) wird das bereits in der Entwurfsphase festgestellte Lösungskonzept mit der Java-Programmiersprache umgesetzt.

Implementierung nach MVC-Modells

Die Implementierung der Applikation erfolgt nach Model-View-Controller-Entwurfsmuster (MVC, Abbildung 4). Das Gesamtsystem wird dementsprechend auf die logischen Komponenten für Datenverarbeitung, Repräsentation und Eingaben geteilt entwickelt.

Erstellung einer MySQL-Datenbank

Anhand der in der Entwurfsphase erstellten ERM- und RM-Modelle wird eine MySQL-Datenbank mit den Tabellen für Regelwerke, Zugfahrten und ZugfahrtDetails mit eindeutigen Attributen und entsprechenden Verknüpfungen zwischen Tabellen erstellt.

Herstellung einer Verbindung mit der Datenbank

Die Herstellung einer Datenbankverbindung erfolgt mithilfe des Hibernate Konfiguration Files (Anhang 10.7), welches alle benötigten Instanzen für eine Verbindungsherstellung mit der Datenbank vordefiniert enthält. Zu den benötigten Instanzen gehören die Konfiguration der Datenbank sowie der Entitäten, deren Zustandsinformationen in der Datenbank gespeichert werden.

Programmierung der Datenmodelle

Die Datenmodelle stellen die Entitäten der Applikation dar und erfassen somit die Kerndaten und -Operationen, um die Daten zur manipulieren. Sie stellen die Zustandsinformationen bereit, damit sie vom Nutzer abgerufen und bei Änderungen aktualisiert werden können.

Programmierung der CRUD-Operationen

Für die Verwaltung der Datenbank werden die CRUD-Operationen (Anhang 10.9) implementiert. Das Akronym CRUD steht für Create, Read, Update und Delete-Methoden, die durchgeführt werden, um die Datensätze in der Datenbank zu speichern, zu ändern, abzurufen sowie zu löschen.

Implementierung der Benutzeroberfläche und Grafiken

Es wurde eine Benutzeroberfläche mit definierten grafischen Komponenten implementiert. Hier ist zu achten, dass die Applikation selbsterklärende und intuitive Bedienung gewährleistet.

Statistische Zugfahrtzeiten

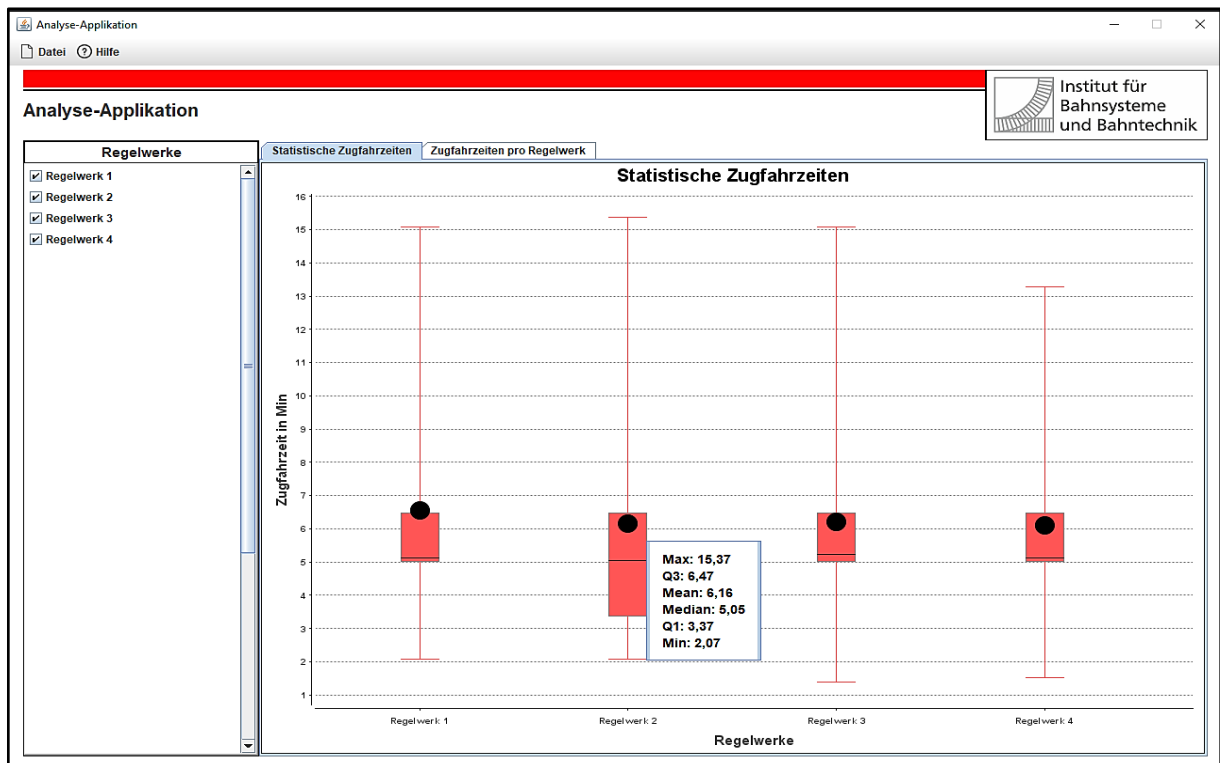


Abbildung 7: Statistische Zugfahrtzeiten (Quelle: Eigene Darstellung)

Zugfahrtzeiten pro Regelwerk

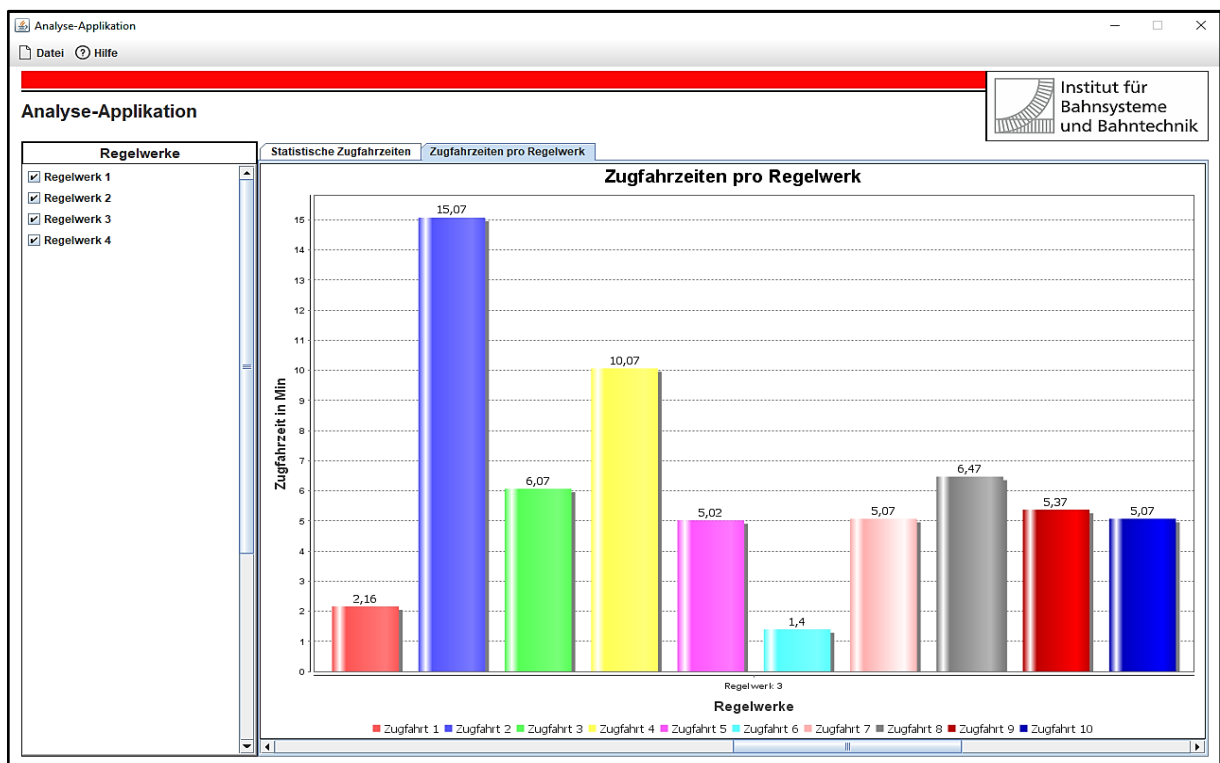


Abbildung 8: Zugfahrtzeiten pro Regelwerk (Quelle: Eigene Darstellung)

Programmierung der Programmlogik

Um das Zusammenspiel der einzelnen Module zu gewährleisten, wird eine Programmlogik bzw. die Steuerungskomponenten programmiert. Diese sind dafür zuständig, dass die durch die Benutzeroberfläche eingehenden Eingaben zu steuern, entsprechend den Datenmodellen modifizieren und anschließend die Ausgabe durch die grafische Oberfläche zurückzuliefern.

3.1.3 Test und Qualitätssicherung

Durchführung eines Funktionstests

Der Funktionstest ist erforderlich, um die Funktionalität der Applikation nach erwartungsgemäßen Verhalten zu prüfen, mögliche Fehler zu entdecken und sie gleich zu beseitigen. Einen detaillierten Testvorgang ist dem Kapitel 3.2 zu entnehmen.

Qualitätssicherung

Die Qualitätssicherung erfolgt durch das Testen und Debugging der Schnittstellen der Applikation während der Implementierungs- sowie Testphasen. Die Implementierung des Projekts nach MVC-Entwurfsmuster schafft ein klares Verständnis des Projektaufbaus und bei Bedarf dessen einfache Weiterentwicklung.

Erstellen eines Testprotokolls

Die zum Testen erforderlichen Testszenarien werden als Messpunkt betrachtet und geprüft, ob alle für das Testszenario unverzichtbaren und verzichtbaren Funktionen zur Verfügung gestellt wurden. Die Bewertung der Testergebnisse erfolgt in Form von einer „Matrix Testauswertung“ mit einer Beschreibung. Eine genauere Darstellung ist dem Kapitel 3.2 zu entnehmen.

3.1.4 Inbetriebnahme

Sobald die im Pflichtenheft beschriebenen Anforderungen erwartungsgemäß umgesetzt und der Funktionstest fehlerfrei abgeschlossen wurde, werden die Projektergebnisse dem Auftraggeber vorgestellt und die benötigten Dokumentationen zur Verfügung gestellt.

3.1.5 Erstellen einer Projektdokumentation

Das Schreiben der Projektdokumentation beginnt in der Planungsphase und wird bis Ende der Projektarbeit vervollständigt. In der Projektdokumentation werden alle geleisteten Prozessschritten detailliert beschrieben.

3.2 Funktionstest

Um die Funktionen der Applikation auf erwartetes und korrektes Verhalten zu prüfen, wird ein Funktionstest nach möglichen Anwendungsfällen, die bereits in der Soll-Ist-Analyse festgestellt wurden, durchgeführt. Die zu erwartenden Ergebnisse nach dem Funktionstest stellen einen umsetzungsfähigen Stand der Applikation dar.

Testprotokoll					
1	Messpunkt	Testaufbau	Anzahl der Testläufe	Bestanden	Gewichtung
2	Applikation starten	Schnittstellen aufrufen	7	Ja	1
3	Datenbankverbindung herstellen	Verbindung zwischen Applikation und Datenbank mittels Hibernate Configuration File herstellen	5	Ja	1
4	Verzeichnis auswählen	Ordner mit Logdateien aus dem Dialogfenster auswählen	4	Ja	1
5	Dateien importieren	Dateien ausschließlich mit „log“-Erweiterung aus dem Verzeichnis importieren	4	Ja	1
6	Datensätze auslesen	Datensätze aus den Dateien auslesen und in einer List speichern	4	Ja	1
7	Datensätze speichern	Session für Transaktion starten, Datensätze in Tabellen speichern, Session beenden	4	Ja	1
8	Datensätze laden	Session für Transaktion starten, Datensätze aus den Tabellen auslesen und in einer Liste speichern, Session beenden, Liste zurückgeben	4	Ja	1
9	Datensätze suchen	Session für Transaktion starten, Datensätze nach bestimmten Kriterien suchen, Session beenden, Datensätze zurückgeben	3	Ja	3
10	Datensätze löschen	Session für Transaktion starten, bestimmte Datensätze aus der Datenbank löschen, Session beenden	3	Ja	2
11	Regelwerke aussuchen	Regelwerke aus der Dropdownliste durch Klick auswählen, Datensätze suchen (9), Grafiken aktualisieren (16)	5	Ja	3
12	Regelwerke löschen	Regelwerke aus der Dropdownliste durch Klick auswählen, Button „Löschen“ drucken, Datensätze löschen (10),	4	Ja	2

		Grafiken aktualisieren (15)			
13	Grafiken anzeigen	Datensätze laden (8) Methode für Auswertung von Datensätzen durchführen Grafiken mit Auswertungsergebnissen erstellen	6	Ja	1
14	Grafiken als PDF speichern	Menüoption „Drucken“ auswählen, Datensätze laden (8), Methode für das Speichern von Grafiken durchführen Option für Speicherortauswahl anzeigen	3	Ja	4
15	Grafiken aktualisieren	Grafiken aus dem GUI löschen, Datensätze laden (8), Grafiken anzeigen	4	Ja	2
16	Statistische Werte anzeigen	Mauszeiger auf grafische Darstellung eines Regelwerks navigieren, Statistische Werte in einem Pop-up-Fenster anzeigen	2	Ja	1
17	Information anfragen	Menüoption „Info“ auswählen, Informationsfenster mit einer Anleitung öffnen	2	Ja	4
18	Applikation beenden	GUI schießen Applikationslauf beenden	5	Ja	1
Legende: Gewichtung ist in Stufen 1 (Unverzichtbar), 2 (Wichtig), 3 (Notwendig), 4 (Möglich), 5 (Verzichtbar) eingestellt.					

Tabelle 7: Matrix Testauswertung (Quelle: Eigene Darstellung)

3.3 Abweichungen und Anpassungen

Es sind keine Abweichungen während der Umsetzung entstanden, daher waren keine Maßnahmen für die Anpassungen erforderlich.

4 Projektergebnisse

4.1 Soll-Ist-Vergleich

Die geplanten Qualitätsmaßnahmen waren effektiv, die Funktionen der Applikation konnten gemäß den möglichen Anwendungsfällen geprüft und bei Bedarf die auftretenden Fehler beseitigt werden.

Es sind keine Abweichungen während der Umsetzung aufgetreten. Das Projekt konnte in den beantragten 70 Zeitstunden erfolgreich abgeschlossen werden.

4.2 Übergabe des Projektergebnisses

Die Projektabnahme erfolgt durch den Auftraggeber nach einer sauberen Durchführung und ordnungsmäßigen Funktionsweise der Analyse-Applikation unter Verwendung von Testdaten. Zusätzlich wird eine Schulungspräsentation über die Anwendungsfunktionalität angeboten und eine Projektdokumentation mit Kundenhandbuch angefertigt übergeben.

4.3 Fazit zum Projekt

Die Projektdurchführung ist gut gelaufen. Währenddessen habe ich viel Neues in Bezug auf die richtige Projektplanung und –implementierung mit der Verwendung des neuen Frameworks JFreeChart Bibliotheken erlernt.

Bezüglich der Projektplanung war es sehr wichtig, die Analysen, Entscheidungen und Planungen hinsichtlich der Projektressourcen ordentlich auszuführen, da diese Schritte bei der Weiterentwicklung von bereits am BST existierten Projekte nicht berücksichtigt werden.

Ebenso war es sinnvoll, die Funktionstests anhand von Testszenarien mehrfach durchzuführen. Dies hat geholfen, die Fehlerursache zu entdecken und gleich zu beseitigen.

Die Zusammenarbeit mit allen Projektbeteiligten fand ich sehr produktiv und wertvoll für das Endergebnis. Sowohl WiMis als auch HiWis waren bereit, bei Rückfragen zu unterstützen und benötigte Informationen zur Verfügung zu stellen.

Für die Projektumsetzung habe ich das Wasserfall-Vorgehensmodell verwendet, da es durch seine klare Struktur einfach einsetzbar ist. Zukünftig werde ich aber agile Vorgehensmodelle verwenden, um die Kommunikation mit dem Auftraggeber zu verbessern, da es während der Übergabe kleine Wünsche bezüglich der grafischen Oberfläche entstanden.

4.4 Ausblick

Die Applikation wird zukünftig weiterentwickelt. Zusätzlich sollen weitere Module realisiert werden, welche die Auswertungen von anderen statistischen Werten mittels verschiedener Grafiken ermöglichen.

5 Abbildungsverzeichnis

Abbildung 1: Stakeholdermatrix. (Quelle: Eigene Darstellung)	5
Abbildung 2: Zeitlicher Projekt-Ablaufplan. (Quelle: Eigene Darstellung. Erstellt mittels des Programms GanttProject 2.8.10 Pilsen unter General Public-License	9
Abbildung 3: Kostenvergleichsrechnung. (Quelle: Eigene Darstellung)	12
Abbildung 4: Architektur der Analyse-Applikation. (Quelle: Eigene Darstellung)	14
Abbildung 5: Anwendungsfalldiagramm der Analyse-Applikation. (Quelle: Eigene Darstellung)	15
Abbildung 6: Datenbankmodell der Analyse-Applikation. (Quelle: Eigene Darstellung)	16
Abbildung 7: Statistische Zugfahrtzeiten (Quelle: Eigene Darstellung)	17
Abbildung 8: Zugfahrtzeiten pro Regelwerk (Quelle: Eigene Darstellung)	17
Abbildung 9: Klassendiagramm der Analyse-Applikation. (Quelle: Eigene Darstellung)	iv
Abbildung 10: Ausschnitt der Inhaltstruktur einer Logdatei (Quelle: Eigene Darstellung)	v
Abbildung 11: Struktur des pom.xml-Files. (Quelle: Eigene Darstellung)	vi
Abbildung 12: Struktur des hibernate.cfg.xml-Files. (Quelle: Eigene Darstellung)	vii
Abbildung 13: Aufbaustruktur der Analyse-Applikation. (Quelle: Eigene Darstellung)	viii

6 Tabellenverzeichnis

Tabelle 1: Stakeholderanalyse. (Quelle: Eigene Darstellung)	4
Tabelle 2: Risikoanalyse. (Quelle: Eigene Darstellung)	7
Tabelle 3: Rollendarstellung aller Personen, die an der Projektdurchführung beteiligt sind. (Quelle: Eigene Darstellung)	8
Tabelle 4: Personalkostenplanung (Quelle: Eigene Darstellung)	10
Tabelle 5: Stundenkosten pro Simulation im Vergleich. (Quelle: Eigene Darstellung)	11
Tabelle 6: Kosten-Nutzen-Analyse. (Quelle: Eigene Darstellung)	12
Tabelle 7: Matrix Testauswertung (Quelle: Eigene Darstellung)	20

7 Literaturverzeichnis

Technische Universität Darmstadt. Eine der führenden Technischen Universitäten. Abgerufen am 17.03.2021 von <https://www.tu-darmstadt.de/universitaet/index.de.jsp>.

Eisenbahnbetriebsfeld Darmstadt. Abgerufen am 13.04.2021 von https://www.verkehr.tu-darmstadt.de/bst/bahnsysteme/einrichtungen_bs/menue-ebd-eisenbahnbetriebsfeld-startseite.de.jsp.

Prof. Dr. Michael H. Breitner, Leibniz Universität Hannover. Vorgehensmodell. Abgerufen am 15.03.2021 von <https://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/is-management/Systementwicklung/Vorgehensmodell/index.html>.

8 Abkürzungsverzeichnis

TU Darmstadt	Technische Universität Darmstadt
HRZ	Hochschulrechenzentrum
BST	Institut für Bahnsysteme und Bahntechnik
EBD	Eisenbahnbetriebsfeld Darmstadt
MVC	Model-View-Controller-Entwurf
SQL	Structured Query Language
ERM	Entity-Relationship-Modell
RM	Relationales Datenbankmodell
JDK	Java Development Kit
CRUD	Create-, Read-, Update- und Delete-Operationen
GNU	GNU General Public License
Eclipse IDE	Eclipse Integrated Development Environment
XAMPP	X steht für alle beliebigen Betriebssystemen, Apache, MySQL, PHP und Perl
WiMi	Wissenschaftliche Mitarbeitende
HiWi	Studentische Hilfskraft
M. Sc.	Master of Science
Dr. rer. nat.	Doctor rerum naturalium (deutsch: Doktor der Naturwissenschaften)
Kap.	Kapitel
bzw.	beziehungsweise
i.d.R.	in der Regel

9 Glossar

smartLogic

Die plattformunabhängige Sicherungslogik für das Eisenbahnstellwerk nächster Generation, die Möglichkeit schafft, die Planungs- und Zulassungsprozesse des Eisenbahnbetriebs zukünftig zu vereinfachen und zu beschleunigen.

Fahrdynamik

Die Fahrdynamik ist ein Gebiet der Dynamik, welche die Ablaufvorgänge bei der Bewegung eines Fahrzeuges beschreibt. Die Kräfte, die auf das Fahrzeug während seiner Fahrt wirken, werden von den Parametern wie Fahrzeuggewicht, Beschleunigung, Bremsen, Luftströmung etc. verursacht.

Regelwerk

Regelwerk ist eine Sammlung von Regeln, nach den Eisenbahnfahrten abgewickelt und Betriebsabläufe gesteuert und überwacht werden.

Logdatei

Eine Logdatei ist eine Datei, in der alle automatisch geführten Prozesse eintragen werden.

MVC

MVC steht für Model-View-Controller, ein Entwurfsmuster, welches ermöglicht, eine Anwendung auf drei wesentlichen Komponenten für die Geschäftslogik, Benutzeroberfläche und Steuerung aufgeteilt zu entwickeln.

MySQL

MySQL ist ein relationales Datenbankverwaltungssystem, das die Operationen zum Speichern, Löschen, Ändern, Abrufen von Datensätzen sowie zum Steuern von Nutzern ausführt.

JFreeChart

JFreeChart ist ein Java basierendes Framework, welche die passenden Bibliotheken für die Erstellung von verschiedenen Diagrammen zur Verfügung stellt.

Hibernate

Hibernate ist ein Java basierendes, plattformunabhängiges Framework, das zur Abbildung von Objekten auf eine relationale Datenbank verwendet wird.

Maven

Maven ist ein Java basierendes Build-Management-Tool, das ermöglicht, die Java-Projekte standardisiert zu erstellen und die Bibliothekabhängigkeiten möglichst einfach zu laden.

Java Development Kit (JDK)

Das Java Development Kit stellt eine Java-Laufzeitumgebung, einen Java-Compiler und die Java-APIs zur Verfügung, um die Java basierenden Anwendungen zu entwickeln.

Eclipse IDE

Eclipse ist eine integrierte Entwicklungsumgebung für die Erstellung von Softwares.

10 Anhang

10.1 Kundendokumentation

Diese Kundendokumentation richtet sich an die Nutzer der Analyse-Applikation und enthält eine ausführliche Information zur deren bestimmungsgemäßen Verwendung.

Mit Hilfe der Analyse-Applikation können beliebige statistischen Werten von Zugfahrtsimulationen bestimmter Regelwerke geladen, in einer Datenbank gespeichert, gelöscht und ausgewählt werden.

10.1.1 Statistische Werte laden

Bitte wählen Sie die Menüoption „Laden“ aus. Dies öffnet ein Dialog zur Auswahl und zum Öffnen von Verzeichnissen. Wählen Sie ein gewünschtes Verzeichnis zum Laden.

Die Applikation liest die Dateien aus dem Verzeichnis, lädt die Datensätze in die Datenbank und stellt eine grafische Auswertung von Datensätzen dem Nutzer bereit.

10.1.2 Statistische Werte löschen

Für das Löschen von statistischen Werten, sollen Sie zuerst ein Haken in Kästchen neben den Regelwerkstiteln setzen, die gelöscht werden sollen, und anschließend die Menüoption „Löschen“ auswählen.

Die Applikation löscht die ausgewählten Datensätze aus der Datenbank und liefert eine neue Instanz der grafischen Auswertung mit aktuellen Datensätzen.

10.1.3 Statistische Werte auswählen

Für die Darstellung von bestimmten statistischen Werten setzen Sie ein Haken in Kästen neben den Regelwerkstiteln, die grafisch ausgewertet werden sollen.

Die Applikation bearbeitet nur die Datensätze von Regelwerken, die abgehakt wurden, und liefert eine neue Instanz der grafischen Auswertung von statistischen Werten der ausgewählten Regelwerke.

10.1.4 Diagramme als PDF speichern

Für das Speichern eines gewünschten Diagramms in einer PDF-Datei, sollen Sie zuerst ein Haken in Kästchen neben den Regelwerken setzen, die in einer PDF-Datei gespeichert werden sollen. und anschließend die Menüoption „als PDF speichern“ auswählen. Dies öffnet ein Dialog zur Auswahl eines Speicherorts. Wählen Sie ein gewünschtes Verzeichnis zum Speichern der PDF-Datei.

10.1.5 Anleitung lesen

Diese Anleitung steht dem Nutzer unter der Menüoption „Info“ zur Verfügung.

10.2 Auszug aus dem Lastenheft

[...]

Ist-Zustand

Die für smartLogic relevanten Fahrregeln werden durch die Simulationen der Zugfahrten im Eisenbahnbetriebsfeld Darmstadt getestet. Es werden die Logdateien von der Fahrdynamik der Modelanlage während der Simulationen erzeugt, die daraufhin von den zuständigen WiMis und HiWis analysiert und bewertet werden. Aufgrund des Fehlens von passenden Werkzeugen wird die Analyse der statistischen Zugfahrtzeiten zurzeit manuell durchgeführt. Da dieser Prozess komplex und aufwändig ist, werden oft Fehler in den Auswertungen nicht erkannt bzw. nicht behoben.

Soll-Konzept

Für eine schnelle und einwandfreie Analyse und Auswertung von statistischen Werten soll eine lauffähige Analyse-Applikation entwickelt werden. Die Applikation soll den Forschungsbeteiligten dabei unterstützen, ein effizientes Szenario mit optimalen Zugfahrtzeiten unter einer getesteten Fahrregel zu ermitteln. Dafür soll die Anwendung eine grafische Auswertung von statistischen Werten bereitstellen. Die Analyse-Applikation soll auf Basis des Model-View-Controller-Entwurfsmusters (MVC) realisiert werden. Dieses ermöglicht eine klare Trennung von programmierten Schnittstellen, welche zukünftig eine Weiterentwicklung der Applikation deutlich vereinfacht. Die Erstellung von Diagrammen soll mit Hilfe des JFreeChart Frameworks erfolgen.

Die Applikation soll daher in der Lage sein, die Logdateien zu lesen, statistische Datensätze in einer Datenbank speichern und sie in Form einer grafischen Auswertung den Nutzer zur Verfügung stellen.

Anforderungen

Funktionalen Anforderungen:

- Logdateien aus dem Pfad wählen und importieren
- Datensätze aus den Logdateien lesen, verarbeiten und mithilfe von Graphiken auswerten
- Datensätze in einer Datenbank speichern und wieder abrufen

Nichtfunktionalen Anforderungen:

- Die Software Architektur soll nach Model-View-Controller-Entwurfsmuster realisiert werden
- Die Graphiken sollen mit Hilfe von JFreeChart Bibliotheken geschaffen werden
- Die Benutzeroberfläche soll selbsterklärend und eine einfache Bedienung gewährleisten

[...]

10.3 Auszug aus dem Pflichtenheft

[...]

Zielbestimmung

In Rahmen des Forschungsprojektes smartLogic soll eine Analyse-Applikation entwickelt werden, welche die statistischen Zugfahrzeiten innerhalb von Testszenarien auswertet, in grafischer Form bereitstellt und in einer Datenbank speichert.

○ **Musskriterien**

- Verzeichnis von Logdateien auswählen
- Logdateien laden
- Datensätze in einer Datenbank speichern
- Datensätze aus der Datenbank laden
- Datensätze nach bestimmten Kriterien suchen
- Datensätze aus der Datenbank löschen
- Statistische Ergebnisse auswerten
- Statistische Ergebnisse grafisch anzeigen

○ **Wunschkriterien**

- Geringere Bearbeitungszeit
- Höhere Auswertungsqualität von statistischen Ergebnissen
- Mehrere Datensätze grafisch auswerten
- Grafische Auswertung als PDF speichern
- Anleitung durch Menüoption zur Verfügung stellen

○ **Abgrenzungskriterien**

- Die Applikation ist nur mit einer MySQL-Datenbank kompatibel.
- Die Applikation verfügt keine Anmeldungsmechanismen
- Es gibt keine Funktion, die die gelöschten Datensätze widerherstellt
- Die Applikation wird ausschließlich auf Deutsch Sprache angeboten, unterstützt keine andere Sprache.

Technische Umgebung

○ **Software**

- Windows 10 Betriebssystem
- Java Virtual Machine

○ **Hardware**

- Rechner
- Peripheriegeräte: Maus, Tastatur, Bildschirm

○ **Orgware**

- Zugangsberechtigung zur Datenbank
- Schulungspräsentation
- Benutzerhandbuch

[...]

10.4 UML-Diagramme

10.4.1 Klassendiagramm

Hierbei wurde ein Abschnitt des Klassendiagramms von Hauptschnittstellen vorgestellt.

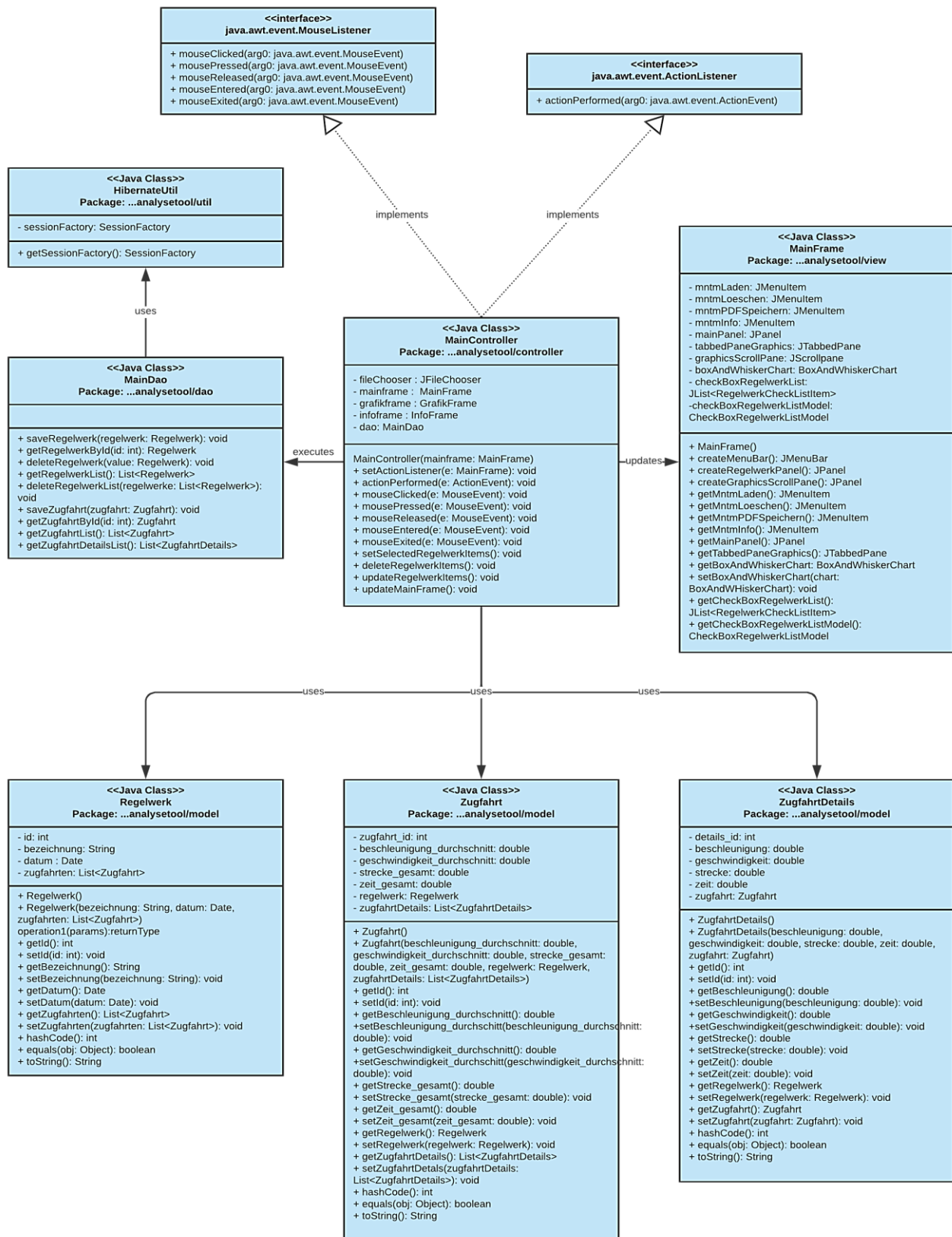


Abbildung 9: Klassendiagramm der Analyse-Applikation. (Quelle: Eigene Darstellung)

10.5 Struktur der Logdatei

[...]

```
[2021-04-21 21:35:25.846] TRN 00001: dd: Set target movement state to: COASTING
[2021-04-21 21:35:25.846] TRN 00001: dd: Acc: 0.00 m/s^2, V: 0.00 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 0.00 m, TripDist: 0.00 m, TripTime: 0.1 s
[2021-04-21 21:35:25.990] TRN 00001: dd: Set target movement state to: ACCELERATING
[2021-04-21 21:35:26.231] TRN 00001: dd: Acc: 0.97 m/s^2, V: 0.12 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 0.01 m, TripDist: 0.01 m, TripTime: 0.5 s
[2021-04-21 21:35:26.352] TRN 00001: dd: Acc: 1.31 m/s^2, V: 0.27 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 0.05 m, TripDist: 0.05 m, TripTime: 0.6 s
[2021-04-21 21:35:26.471] TRN 00001: dd: Acc: 1.66 m/s^2, V: 0.47 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 0.10 m, TripDist: 0.10 m, TripTime: 0.7 s
[2021-04-21 21:35:26.571] TRN 00001: dd: Acc: 2.00 m/s^2, V: 0.67 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 0.17 m, TripDist: 0.17 m, TripTime: 0.8 s
[2021-04-21 21:35:26.672] TRN 00001: dd: Acc: 2.34 m/s^2, V: 0.90 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 0.26 m, TripDist: 0.26 m, TripTime: 0.9 s
[2021-04-21 21:35:27.623] TRN 00001: dd: Acc: 3.70 m/s^2, V: 4.21 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 2.79 m, TripDist: 2.79 m, TripTime: 1.9 s
[2021-04-21 21:35:28.576] TRN 00001: dd: Acc: 3.70 m/s^2, V: 7.73 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 8.65 m, TripDist: 8.65 m, TripTime: 2.8 s
[2021-04-21 21:35:29.538] TRN 00001: dd: Acc: 3.70 m/s^2, V: 11.27 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 17.96 m, TripDist: 17.96 m, TripTime: 3.8 s
[2021-04-21 21:35:30.453] TRN 00001: dd: Acc: 3.69 m/s^2, V: 14.65 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 29.98 m, TripDist: 29.98 m, TripTime: 4.7 s
[2021-04-21 21:35:31.401] TRN 00001: dd: Acc: 3.69 m/s^2, V: 18.14 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 45.68 m, TripDist: 45.68 m, TripTime: 5.7 s
[2021-04-21 21:35:32.334] TRN 00001: dd: Acc: 3.59 m/s^2, V: 21.56 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 64.36 m, TripDist: 64.36 m, TripTime: 6.6 s
[2021-04-21 21:35:33.297] TRN 00001: dd: Acc: 3.10 m/s^2, V: 24.73 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 86.81 m, TripDist: 86.81 m, TripTime: 7.6 s
[2021-04-21 21:35:34.231] TRN 00001: dd: Acc: 2.77 m/s^2, V: 27.45 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 111.31 m, TripDist: 111.31 m, TripTime: 8.5 s
[2021-04-21 21:35:35.142] TRN 00001: dd: Acc: 2.53 m/s^2, V: 29.84 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 137.50 m, TripDist: 137.50 m, TripTime: 9.4 s
[2021-04-21 21:35:36.066] TRN 00001: dd: Acc: 2.34 m/s^2, V: 32.07 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 166.16 m, TripDist: 166.16 m, TripTime: 10.3 s
[2021-04-21 21:35:36.977] TRN 00001: dd: Acc: 2.10 m/s^2, V: 34.11 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 196.40 m, TripDist: 196.40 m, TripTime: 11.2 s
[2021-04-21 21:35:37.888] TRN 00001: dd: Acc: 1.63 m/s^2, V: 35.78 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 228.30 m, TripDist: 228.30 m, TripTime: 12.1 s
[2021-04-21 21:35:38.798] TRN 00001: dd: Acc: 1.56 m/s^2, V: 37.23 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 261.55 m, TripDist: 261.55 m, TripTime: 13.0 s
[2021-04-21 21:35:39.710] TRN 00001: dd: Acc: 1.50 m/s^2, V: 38.62 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 296.21 m, TripDist: 296.21 m, TripTime: 14.0 s
[2021-04-21 21:35:40.639] TRN 00001: dd: Acc: 1.44 m/s^2, V: 39.98 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 332.74 m, TripDist: 332.74 m, TripTime: 14.9 s
[2021-04-21 21:35:41.594] TRN 00001: dd: Acc: 1.02 m/s^2, V: 41.17 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 371.55 m, TripDist: 371.55 m, TripTime: 15.8 s
[2021-04-21 21:35:42.513] TRN 00001: dd: Acc: 0.80 m/s^2, V: 41.94 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 409.73 m, TripDist: 409.73 m, TripTime: 16.8 s
[2021-04-21 21:35:43.475] TRN 00001: dd: Acc: 0.78 m/s^2, V: 42.69 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 450.47 m, TripDist: 450.47 m, TripTime: 17.7 s
[2021-04-21 21:35:44.415] TRN 00001: dd: Acc: 0.66 m/s^2, V: 43.40 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 490.93 m, TripDist: 490.93 m, TripTime: 18.7 s
[2021-04-21 21:35:45.342] TRN 00001: dd: Acc: 0.22 m/s^2, V: 43.79 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 531.34 m, TripDist: 531.34 m, TripTime: 19.6 s
[2021-04-21 21:35:46.264] TRN 00001: dd: Acc: 0.22 m/s^2, V: 43.99 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 571.73 m, TripDist: 571.73 m, TripTime: 20.5 s
[2021-04-21 21:35:47.175] TRN 00001: dd: Acc: 0.05 m/s^2, V: 44.10 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 611.80 m, TripDist: 611.80 m, TripTime: 21.4 s
[2021-04-21 21:35:48.088] TRN 00001: dd: Acc: 0.05 m/s^2, V: 44.15 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 651.97 m, TripDist: 651.97 m, TripTime: 22.3 s
[2021-04-21 21:35:49.018] TRN 00001: dd: Acc: 0.05 m/s^2, V: 44.20 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 692.72 m, TripDist: 692.72 m, TripTime: 23.2 s
[2021-04-21 21:35:49.928] TRN 00001: dd: Acc: 0.05 m/s^2, V: 44.25 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 732.92 m, TripDist: 732.92 m, TripTime: 24.2 s
[2021-04-21 21:35:50.837] TRN 00001: dd: Acc: 0.05 m/s^2, V: 44.30 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 773.10 m, TripDist: 773.10 m, TripTime: 25.1 s
[2021-04-21 21:35:51.747] TRN 00001: dd: Acc: 0.05 m/s^2, V: 44.35 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 813.38 m, TripDist: 813.38 m, TripTime: 26.0 s
[2021-04-21 21:35:52.657] TRN 00001: dd: Acc: 0.05 m/s^2, V: 44.39 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 853.69 m, TripDist: 853.69 m, TripTime: 26.9 s
[2021-04-21 21:35:53.569] TRN 00001: dd: Acc: 0.05 m/s^2, V: 44.44 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 894.19 m, TripDist: 894.19 m, TripTime: 27.8 s
[2021-04-21 21:35:53.771] TRN 00001: sds: Current speed supervision intervention level: OVERSPEED
[2021-04-21 21:35:54.479] TRN 00001: dd: Acc: 0.05 m/s^2, V: 44.49 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 934.62 m, TripDist: 934.62 m, TripTime: 28.7 s
[2021-04-21 21:35:55.389] TRN 00001: dd: Acc: 0.05 m/s^2, V: 44.53 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 975.07 m, TripDist: 975.07 m, TripTime: 29.6 s
[2021-04-21 21:35:56.301] TRN 00001: dd: Acc: 0.05 m/s^2, V: 44.58 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 1015.65 m, TripDist: 1015.65 m, TripTime: 30.5 s
[2021-04-21 21:35:57.212] TRN 00001: dd: Acc: 0.05 m/s^2, V: 44.62 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 1056.24 m, TripDist: 1056.24 m, TripTime: 31.4 s
[2021-04-21 21:35:58.124] TRN 00001: dd: Acc: 0.05 m/s^2, V: 44.67 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 1096.93 m, TripDist: 1096.93 m, TripTime: 32.3 s
[2021-04-21 21:35:59.034] TRN 00001: dd: Acc: 0.05 m/s^2, V: 44.72 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 1137.54 m, TripDist: 1137.54 m, TripTime: 33.3 s
[2021-04-21 21:35:59.942] TRN 00001: dd: Acc: 0.05 m/s^2, V: 44.76 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 1178.16 m, TripDist: 1178.16 m, TripTime: 34.2 s
[2021-04-21 21:36:00.912] TRN 00001: sds: Current speed supervision state: TARGET_SPEED_SUPERVISION
[2021-04-21 21:36:00.912] TRN 00001: dd: Acc: 0.05 m/s^2, V: 44.81 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 1221.55 m, TripDist: 1221.55 m, TripTime: 35.1 s
[2021-04-21 21:36:01.815] TRN 00001: dd: Acc: 0.05 m/s^2, V: 44.85 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 1261.96 m, TripDist: 1261.96 m, TripTime: 36.0 s
[2021-04-21 21:36:02.734] TRN 00001: dd: Set target movement state to: COASTING
[2021-04-21 21:36:02.734] TRN 00001: dd: Acc: 0.05 m/s^2, V: 44.89 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 1303.14 m, TripDist: 1303.14 m, TripTime: 36.9 s
[2021-04-21 21:36:03.672] TRN 00001: dd: Acc: -0.14 m/s^2, V: 44.76 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 1345.17 m, TripDist: 1345.17 m, TripTime: 37.9 s
[2021-04-21 21:36:04.590] TRN 00001: dd: Acc: -0.14 m/s^2, V: 44.63 m/s, V_max: 44.44 m/s, Pos: LRBG 1+ 1386.14 m, TripDist: 1386.14 m, TripTime: 38.8 s
[2021-04-21 21:36:05.509] TRN 00001: dd: Acc: -0.14 m/s^2, V: 44.51 m/s, V_max: 44.25 m/s, Pos: LRBG 1+ 1427.03 m, TripDist: 1427.03 m, TripTime: 39.7 s
[2021-04-21 21:36:06.325] TRN 00001: dd: Set target movement state to: SERVICE_BREAKING
[2021-04-21 21:36:06.325] TRN 00001: dd: Acc: -0.14 m/s^2, V: 44.38 m/s, V_max: 43.60 m/s, Pos: LRBG 1+ 1467.74 m, TripDist: 1467.74 m, TripTime: 40.6 s
```

[...]

Abbildung 10: Ausschnitt der Inhaltstruktur einer Logdatei (Quelle: Eigene Darstellung)

10.6 POM-Struktur

```
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>de.tu-darmstadt</groupId>
  <artifactId>analysetool</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <properties>
    <maven.compiler.source>15</maven.compiler.source>
    <maven.compiler.target>15</maven.compiler.target>
  </properties>

  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-jar-plugin</artifactId>
        <configuration>
          <descriptorRefs>
            <descriptorRef>jar-with-dependencies</descriptorRef>
          </descriptorRefs>
          <archive>
            <manifest>
              <addClasspath>true</addClasspath>
              <mainClass>de.tudarmstadt.fgbs.analysetool.main.Main</mainClass>
            </manifest>
          </archive>
        </configuration>
      </plugin>
    </plugins>
  </build>

  <dependencies>
    <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>8.0.22</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
    <dependency>
      <groupId>org.hibernate</groupId>
      <artifactId>hibernate-core</artifactId>
      <version>5.4.27.Final</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/jfree/jfreechart -->
    <dependency>
      <groupId>jfree</groupId>
      <artifactId>jfreechart</artifactId>
      <version>1.0.13</version>
    </dependency>

    <!-- https://mvnrepository.com/artifact/com.lowagie/itext -->
    <dependency>
      <groupId>com.lowagie</groupId>
      <artifactId>itext</artifactId>
      <version>2.1.7</version>
    </dependency>
  </dependencies>

</project>
```

Abbildung 11: Struktur des pom.xml-Files. (Quelle: Eigene Darstellung)

10.7 Hibernate Konfiguration File

```
<?xmlversion="1.0"encoding="UTF-8"?>
<!DOCTYPEhibernate-configurationPUBLIC
"-//Hibernate/Hibernate Configuration DTD 3.0//EN"
"http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">

<hibernate-configuration>
  <session-factory>
    <!-- JDBC Database connection settings -->
    <propertyname="connection.driver_class">com.mysql.jdbc.Driver</property>
    <propertyname="connection.url">_____</property>
    <propertyname="connection.username">_____</property>
    <propertyname="connection.password">_____</property>

    <propertyname="hibernate.hbm2ddl.auto">update</property>

    <!-- Select our SQL dialect -->
    <propertyname="dialect">org.hibernate.dialect.MySQL5Dialect</property>

    <!-- JDBC connection pool settings ... using built-in test pool -->
    <propertyname="connection.pool_size">10</property>

    <!-- Disable the second-level cache -->
    <propertyname="cache.provider_class">org.hibernate.cache.NoCacheProvider</property>

    <!-- Enable Hibernate's automatic session context management -->
    <propertyname="current_session_context_class">thread</property>
    <propertyname="hibernate.enable_lazy_load_no_trans">true</property>

    <!-- Mapping files -->
    <mapping class="de.tudarmstadt.fgbs.analysetool.model.Regelwerk"></mapping>
    <mapping class="de.tudarmstadt.fgbs.analysetool.model.ZugfahrtProRegelwerk"></mapping>
    <mapping class="de.tudarmstadt.fgbs.analysetool.model.ZugfahrtDetails"></mapping>

  </session-
</hibernate-configuration>>
```

Abbildung 12: Struktur des hibernate.cfg.xml-Files.¹ (Quelle: Eigene Darstellung)

¹ Aus Sicherheitsgründen wurden das Datenbankverzeichnis, der Benutzername sowie das Passwort entfernt.

10.8 Struktur

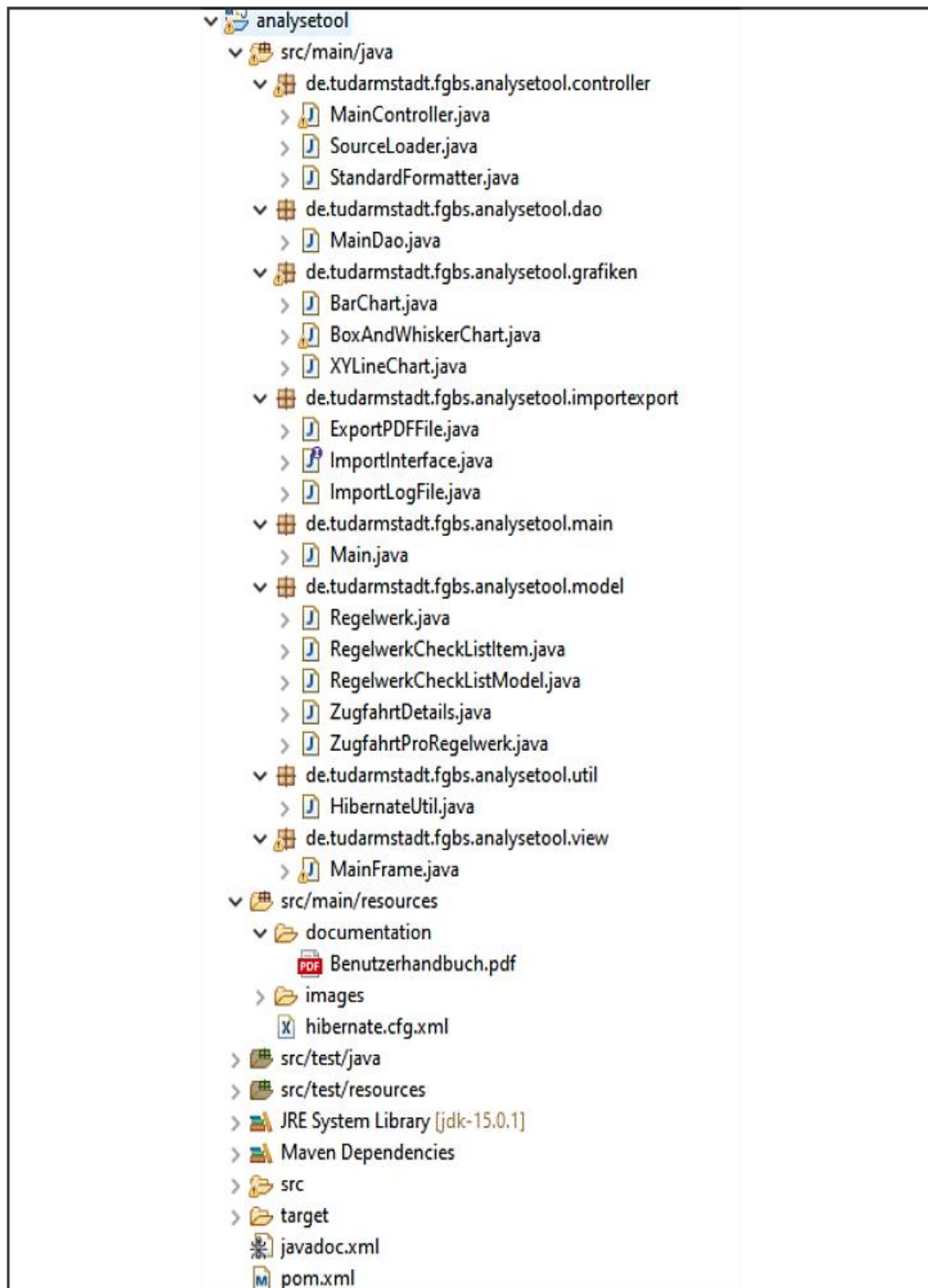


Abbildung 13: Aufbaustruktur der Analyse-Applikation. (Quelle: Eigene Darstellung)

10.9 Auszug aus dem Quellcode

```
1  /*****
2  *                               HibernateUtil
3  *****/
4  package de.tudarmstadt.fgbs.analysetool.util;
5
6  import org.hibernate.SessionFactory;
7  import org.hibernate.cfg.Configuration;
8
9  public class HibernateUtil {
10
11     private static SessionFactory sessionFactory;
12
13     /**
14     * baut ein SessionFactory einmal pro Anwendung auf
15     * SessionFactory lädt die Konfiguration und die Abbildungen aus dem
16     * 'hibernate.cfg.xml'-File, um mit der Datenbank zu verbinden und
17     * die CRUD-Operationen auszuführen
18     * @return sessionFactory
19     */
20
21     public static SessionFactory getSessionFactory() {
22
23         if(sessionFactory == null) {
24             try {
25                 sessionFactory =
26                     new Configuration().configure("hibernate.cfg.xml").
27                     buildSessionFactory();
28
29             } catch (Exception e) {
30                 e.printStackTrace();
31             }
32         }
33         return sessionFactory;
34     }
35 }
36
37 /*****
38 *                               Model Regelwerk
39 *****/
40 package de.tudarmstadt.fgbs.analysetool.model;
41
42 import java.io.Serializable;
43 import java.util.Date;
44 import java.util.List;
45 import javax.persistence.CascadeType;
46 import javax.persistence.Column;
47 import javax.persistence.Entity;
48 import javax.persistence.FetchType;
49 import javax.persistence.GeneratedValue;
50 import javax.persistence.GenerationType;
51 import javax.persistence.Id;
52 import javax.persistence.OneToOne;
53 import javax.persistence.Table;
54
55 /**
56 * Entitätsklasse Regelwerk für die Datenbanktabelle 'regelwerk'.
57 * enthält die Attributen 'regelwerk_id' (Primärschlüssel),
58 * 'bezeichnung', 'geschwindigkeit', 'datum'.
59 * @author tashkulova
60 *
61 */
62
63 @Entity
64 @Table(name = "regelwerk")
```

```

65 public class Regelwerk implements Serializable{
66     /**
67      *
68      */
69     private static final long serialVersionUID = 1L;
70
71     @Id
72     @GeneratedValue(strategy = GenerationType.AUTO)
73     @Column(name = "regelwerk_id", nullable = false)
74     private int id;
75
76     @Column(name = "bezeichnung", nullable = false, unique = true)
77     private String bezeichnung;
78
79     @Column(name = "datum", nullable = false)
80     private java.util.Date datum;
81
82
83     @OneToMany(fetch=FetchType.EAGER, cascade={CascadeType.ALL},
84             mappedBy = "regelwerk", orphanRemoval = true)
85     private List<ZugfahrtProRegelwerk> zugfahrten;
86
87     public Regelwerk() {
88         super();
89     }
90
91     public Regelwerk(String bezeichnung, Date datum,
92                     List<ZugfahrtProRegelwerk> zugfahrten) {
93         super();
94         this.bezeichnung = bezeichnung;
95         this.datum = datum;
96         this.zugfahrten = zugfahrten;
97     }
98
99     public int getId() {
100         return id;
101     }
102
103     public void setId(int id) {
104         this.id = id;
105     }
106
107     public String getBezeichnung() {
108         return bezeichnung;
109     }
110
111     public void setBezeichnung(String bezeichnung) {
112         this.bezeichnung = bezeichnung;
113     }
114
115     public java.util.Date getDatum() {
116         return datum;
117     }
118
119     public void setDatum(java.util.Date datum) {
120         this.datum = datum;
121     }
122
123     public List<ZugfahrtProRegelwerk> getZugfahrten() {
124         return zugfahrten;
125     }
126
127     public void setZugfahrten(List<ZugfahrtProRegelwerk> zugfahrten) {
128         this.zugfahrten = zugfahrten;
129     }
130
131     public static long getSerialversionuid() {
132         return serialVersionUID;

```

```

133     }
134
135     @Override
136     public int hashCode() {
137         final int prime = 31;
138         int result = 1;
139         result = prime * result + id;
140         return result;
141     }
142
143     @Override
144     public boolean equals(Object obj) {
145         if (this == obj)
146             return true;
147         if (obj == null)
148             return false;
149         if (getClass() != obj.getClass())
150             return false;
151         Regelwerk other = (Regelwerk) obj;
152         if (id != other.id)
153             return false;
154         return true;
155     }
156
157     @Override
158     public String toString() {
159         return "Regelwerk [bezeichnung=" + bezeichnung + ",
160                                     datum=" + datum + "]\n";
161     }
162 }
163
164
165 /*****
166 *                               MainDao
167 *****/
168 package de.tudarmstadt.fgbs.analysetool.dao;
169
170 import java.util.List;
171 import org.hibernate.Session;
172 import org.hibernate.Transaction;
173 import de.tudarmstadt.fgbs.analysetool.model.Regelwerk;
174 import de.tudarmstadt.fgbs.analysetool.model.ZugfahrtProRegelwerk;
175 import de.tudarmstadt.fgbs.analysetool.model.ZugfahrtDetails;
176 import de.tudarmstadt.fgbs.analysetool.util.HibernateUtil;
177
178 /**
179  * Klasse 'MainDao' (Data Access Object) dient zum Ausführen von
180  * 'CRUD'-Operationen (Create, Read, Update und Delete-Operationen)
181  * @author tashkulova
182  *
183  */
184
185 public class MainDao {
186
187     public MainDao() {
188     }
189
190     /**
191      * speichert ein Objekt der Klasse 'Regelwerk' in der Tabelle
192      * 'regelwerk'
193      * @param regelwerk
194      */
195
196     public void saveRegelwerk(Regelwerk regelwerk) {
197
198         Session session = null;
199         Transaction transaction = null;
200
201         try {

```

```

202         session = HibernateUtil.getSessionFactory().openSession();
203         transaction = session.beginTransaction();
204
205         if(regelwerk.getId() == 0) {
206             session.persist(regelwerk);
207         }else {
208             session.merge(regelwerk);
209         }
210         transaction.commit();
211     }catch(Exception e) {
212         if(transaction != null) {
213             transaction.rollback();
214         }
215     }
216     finally {
217         session.close();
218     }
219 }
220
221 /**
222  * holt ein Objekt der Klasse 'Regelwerk' mithilfe eines 'id' Schlüssels
223  * aus der Tabelle 'regelwerk' raus
224  * @param id
225  * @return Regelwerk
226  */
227
228 public Regelwerk getRegelwerkById(int id) {
229
230     Session session = null;
231     Transaction transaction = null;
232     Regelwerk regelwerk = null;
233
234     try {
235         session = HibernateUtil.getSessionFactory().openSession();
236         transaction = session.beginTransaction();
237         regelwerk = session.get(Regelwerk.class, id);
238         transaction.commit();
239     }catch(Exception e) {
240         if(transaction != null) {
241             transaction.rollback();
242         }
243     }
244     finally {
245         session.close();
246     }
247     return regelwerk;
248 }
249
250 /**
251  * löscht ein Objekt der Klasse 'Regelwerk' aus der
252  * Tabelle 'regelwerk'
253  * @param value
254  */
255
256 public void deleteRegelwerk(Regelwerk value) {
257
258     Session session = null;
259     Transaction transaction = null;
260     Regelwerk regelwerk = null;
261
262     try {
263         session = HibernateUtil.getSessionFactory().openSession();
264         transaction = session.beginTransaction();
265         regelwerk = session.get(Regelwerk.class, value.getId());
266         session.delete(regelwerk);
267         transaction.commit();
268     }catch(Exception e) {
269         if(transaction != null) {
270             transaction.rollback();

```

```

271     }
272     }finally {
273         session.close();
274     }
275 }
276
277 /**
278  * holt eine List von Objekten der Klasse 'Regelwerk' aus der Tabelle
279  * 'regelwerk' raus
280  * @return List<Regelwerk>
281  */
282
283 @SuppressWarnings("unchecked")
284 public List<Regelwerk> getRegelwerkList(){
285
286     Session session = null;
287     Transaction transaction = null;
288     List<Regelwerk> regelwerkList = null;
289
290     try {
291         session = HibernateUtil.getSessionFactory().openSession();
292         transaction = session.beginTransaction();
293         regelwerkList = session.createQuery("FROM Regelwerk "+
294             "ORDER BY regelwerk_id ASC").getResultList();
295         transaction.commit();
296     }catch(Exception e) {
297         if(transaction != null) {
298             transaction.rollback();
299         }
300     }finally {
301         session.close();
302     }
303     return regelwerkList;
304 }
305
306 /**
307  * löscht alle Objekte der Klasse 'Regelwerk', die im Parameter
308  * 'regelwerke' enthalten sind, aus der Tabelle 'regelwerk'
309  * @param regelwerke
310  */
311
312 public void deleteRegelwerkList(List<Regelwerk> regelwerke) {
313
314     Session session = null;
315     Transaction transaction = null;
316
317     try {
318         session = HibernateUtil.getSessionFactory().openSession();
319         transaction = session.beginTransaction();
320         for(Regelwerk regelwerk : regelwerke) {
321             regelwerk = session.get(Regelwerk.class, regelwerk.getId());
322             session.delete(regelwerk);
323         }
324         transaction.commit();
325     }catch(Exception e) {
326         if(transaction != null) {
327             transaction.rollback();
328         }
329     }finally {
330         session.close();
331     }
332 }
333
334 /**
335  * speichert ein Objekt der Klasse 'ZugfahrtProRegelwerk' in der
336  * Tabelle 'zugfahrt'
337  * @param zugfahrtProRegelwerk
338  */
339

```

```

340     public void saveZugfahrt(ZugfahrtProRegelwerk zugfahrtProRegelwerk) {
341
342         Session session = null;
343         Transaction transaction = null;
344
345         try {
346             session = HibernateUtil.getSessionFactory().openSession();
347             transaction = session.beginTransaction();
348             if(zugfahrtProRegelwerk.getId() == 0) {
349                 session.persist(zugfahrtProRegelwerk);
350             }else {
351                 session.merge(zugfahrtProRegelwerk);
352             }
353             transaction.commit();
354         }catch(Exception e) {
355             if(transaction != null) {
356                 transaction.rollback();
357             }
358         }finally {
359             session.close();
360         }
361     }
362
363     /**
364     * holt ein Objekt der Klasse 'ZugfahrtProRegelwerk' mithilfe eines
365     * 'id' Schlüssels aus der Tabelle 'zugfahrt' raus
366     * @param id
367     * @return Zugfahrt
368     */
369
370     public ZugfahrtProRegelwerk getZugfahrtById(int id) {
371
372         Session session = null;
373         Transaction transaction = null;
374         ZugfahrtProRegelwerk zugfahrtProRegelwerk = null;
375
376         try {
377             session = HibernateUtil.getSessionFactory().openSession();
378             transaction = session.beginTransaction();
379
380             zugfahrtProRegelwerk = session.get(ZugfahrtProRegelwerk.class, id);
381         }catch(Exception e) {
382             if(transaction != null) {
383                 transaction.rollback();
384             }
385         }finally {
386             session.close();
387         }
388         return zugfahrtProRegelwerk;
389     }
390
391
392     /**
393     * holt eine Liste von Objekten der Klasse 'ZugfahrtProRegelwerk'
394     * aus der Tabelle 'zugfahrt' raus
395     * @return List<Zugfahrt>
396     */
397
398     @SuppressWarnings("unchecked")
399     public List<ZugfahrtProRegelwerk> getZugfahrtList(){
400
401         Session session = null;
402         Transaction transaction = null;
403         List<ZugfahrtProRegelwerk> zugfahrtList = null;
404
405         try {
406             session = HibernateUtil.getSessionFactory().openSession();
407             transaction = session.beginTransaction();
408             zugfahrtList = session.createQuery("FROM ZugfahrtProRegelwerk "+

```



```

409         "ORDER BY zugfahrt_id ASC").getResultList();
410         transaction.commit();
411     }catch(Exception e) {
412         if(transaction != null) {
413             transaction.rollback();
414         }
415     }finally {
416         session.close();
417     }
418     return zugfahrtList;
419 }
420
421 /**
422  * holt eine Liste von Objekten der Klasse 'ZugfahrtDetails' aus der
423  * Tabelle 'zugfahrt_details' raus
424  * @return List<ZugfahrtDetails>
425  */
426 @SuppressWarnings("unchecked")
427 public List<ZugfahrtDetails> getZugfahrtDetailsList() {
428
429     Session session = null;
430     Transaction transaction = null;
431     List<ZugfahrtDetails> zugfahrtDetailsList = null;
432
433     try {
434         session = HibernateUtil.getSessionFactory().openSession();
435         transaction = session.beginTransaction();
436         zugfahrtDetailsList = session.createQuery("FROM ZugfahrtDetails "+
437             "ORDER BY details_id ASC").getResultList();
438         transaction.commit();
439     }catch(Exception e) {
440         if(transaction != null) {
441             transaction.rollback();
442         }
443     }finally {
444         session.close();
445     }
446
447     return zugfahrtDetailsList;
448 }
449
450 }

```