Entwicklungsoptionen zur Unterstützung von Brückeninspektionen mit mobilen Geräten

Diplomarbeit

Ausgeführt zum Zweck der Erlangung des akademischen Grades
Dipl.-Ing. für technisch-wissenschaftliche Berufe

am Masterstudiengang Digitale Medientechnologien an der
Fachhochschule St. Pölten, **Masterklasse Mobile**

von:
Gerald Strobl, BSc
dm151534

Betreuerin und Erstbegutachterin: DI Kerstin Blumenstein, BSc
Zweitbegutachterin: Dipl.-Ing. Dipl.-Ing. Dr. Hirut Grossberger

St.Pölten, 18.11.2017

Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.

- ich dieses Thema bisher weder im Inland noch im Ausland einem Begutachter oder einer Begutachterin zur Beurteilung, oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Diese Arbeit stimmt mit der vom Begutachter bzw. der Begutachterin beurteilten Arbeit überein.

....

Ort, Datum Unterschrift

Kurzfassung

Im 21. Jahrhundert werden im täglichen Leben immer mehr Bereiche und Prozesse durch die Digitalisierung von Tablets und Smartphones vereinfacht. Applikationen sind allgegenwärtig und teilweise kaum noch wegzudenken. Es gibt aber noch Bereiche, die zur Gänze auf analoge Methoden zugreifen. Im technischen Bauwesen werden Brückeninspektionen noch klassisch mit Papier, Stift und Klemmbrett abgewickelt.

In dieser Diplomarbeit wird untersucht, wie der Brückeninspektionsprozess mit Tablets oder Smartphones verbessert werden kann.

Konzepte und Ansätze bestehender Forschungsarbeiten sowie Konkurrenzanalysen werden durch eine Literaturrecherche untersucht. Durch ein Expertinnen Interview und einer Problemanalyse wird der Prozess der Brückeninspektion bis ins Detail erforscht. Zusätzlich werden Analysen im Bereich der mobilen Entwicklungstechnologien durchgeführt, um den neusten Stand der Technik in diesem Bereich zu untersuchen.

Durch die Problemanalyse wird ein Anforderungsprofil erstellt und mit der technischen Komponente verknüpft. Dadurch entstehen die Rahmenbedingungen für einen Prototyp in Form einer Applikation. Die App wird im Rahmen der Diplomarbeit mit User Tests untersucht und evaluiert. Durch die Ergebnisse der Evaluierung können dann Rückschlüsse auf die Funktionalität, Usability und mögliche Prozessoptimierung der Applikation gezogen werden.

Abstract

In the 21st century, the digitalization of smartphones and tablets is known to simplify more and more areas and processes of everyday life. Applications are all around us an often times they are thought of being indispensable. Although the technology has come so far, in some areas analog processes are still relevant. For instance, in the technical construction business bridge inspections are still conducted with pencil and paper.

The topic of this thesis is how smartphones and tablet can improve bridge inspection processes.

In this thesis, concepts and different approaches in existing research as well as competitive analyses are examined in a literary research. Additionally, there are researches about other applications for bridge inspections. To take a closer look at the details of a bridge inspection process an expert interview and a problem analysis have been implemented. In addition, a technical evolution on application development has been included in order to the get an overview on trends, advantages and disadvantages. Also analyses regarding mobile developing technologies are examined in order to analyse the state-of-the-art of this new technology.

By means of this analysis a requirement profile is provided and then connected with the technical aspects. This is how the parameters for a prototype in form of an application are defined. Another important point is the user tests. The user tests are implemented in the thesis to evaluate the prototype. The evaluation results are the primary factor to get a conclusion regarding the functionality, usability and a possible process optimization of the application.

Inhaltsverzeichnis

Ehrenwörtliche Erklärung II

Kurzfassung III

Abstract IV

Inhaltsverzeichnis V

1 Einleitung 8

1.1 Problemstellung und Relevanz 8

1.2 Ziele der Arbeit 9

1.3 Forschungsfragen und Hypothesen 10

1.4 Methodik und Aufbau der Arbeit 11

1.5 Definitionen 12

1.5.1 Mobile Entwicklungstechnologien 12

1.5.2 Container 12

1.5.3 MVC 12

1.5.4 Routing 12

1.5.5 API Schnittstelle 13

2 Theoretische Ausgangslage 14

2.1 Forschungsprojekte 14

2.1.1 Research and Application of Bridge Inspection and Maintenance Management System 14

2.1.2 An effective practices, strategies and technologies in the service industry to increase customer loyalty using map indicator 19

2.2 Erkenntnisse der Forschungsprojekte 19

2.3 Konkurrenzanalyse: SIB-BAUWERKE 20

3 Mobile Entwicklungstechnologien 22

3.1 Webbasierte Entwicklungstechnologien 22

3.1.1 Ionic 24

3.1.2 Meteor 25

3.1.3 Sencha Touch/Ext JS 26

3.1.4 Frontend JS Bibliotheken 27

3.1.5 Progressive Web Apps 30

3.1.6 Funktionen und Fazit von Web Applikationen 32

3.2 Hybride Entwicklungstechnologien 33

3.2.1 PhoneGap/Cordova 35

3.2.2 Adobe Air 36

3.2.3 React Native 37

3.2.4 Appcelerator Titanium 37
3.2.5 Funktionen und Vergleich der Frameworks 38
3.3 Native Entwicklungstechnologien 38
3.3.1 Eigenschaften und Funktionalität einer nativen Applikation 39
3.3.2 iOS Betriebssystem 40
3.3.3 Android Betriebssystem 41
3.3.4 Interface Guidelines Vergleich iOS und Android 41
3.3.5 Fazit 43
3.4 Vergleich der Entwicklungsmethoden 43

4 Problemanalyse 47
4.1 Expertinnen Interview 47
4.1.1 Leitfaden 47
4.1.2 Durchführung 47
4.1.3 Anforderungserhebung 48
4.2 Anforderungserhebung Dokumentenanalyse 51
4.2.1 Ansatz der Analyse 52
4.2.2 Dokument ÖBB: Richtlinie für die Überwachung und Inspektion von Eisenbahnbrücken in Spannbetonweise 52
4.2.3 Dokument Bundesministerium für Verkehr, Bau und Stadtentwicklung: Bauwerksprüfung nach DIN 1076 Bedeutung, Organisation, Kosten 63
4.2.4 Dokument ÖBB: Revisionsblatt 66
4.3 Zusammenfassung 70

5 Anforderungen und technische Entscheidungen 71
5.1 Anforderungsprofil 71
5.1.1 Allgemeine Anforderungen 71
5.1.2 Brückenverwaltung 72
5.1.3 Inspektionsverwaltung 72
5.1.4 Dokumentationsmöglichkeiten 73
5.1.5 Dokumentation von Schäden 74
5.1.6 Archivierungsverwaltung 74
5.1.7 Benutzerverwaltung und Sicherheit 75
5.2 Technische Umsetzung 75
5.2.1 Mobile Entwicklungstechnologien 75
5.2.2 Systemarchitektur 76
5.2.3 Datenbanksystem 77

6 Prototyp 79
6.1 Technik 79
6.1.1 Backend Technologie 79
6.1.2 Frontend Technologie 79

6.1.3 API Schnittstelle 80
6.2 Umsetzung 80
6.2.1 Lokale Installation 80
6.2.2 Setup von Git 81
6.2.3 Implementierung von Vue 81
6.2.4 Konfiguration einer Single-Page-Webanwendung 82
6.2.5 Dateistruktur der Applikation 83
6.2.6 Datenbankstruktur 83
6.2.7 Routing 85
6.2.8 Methodik der Templates 87
6.2.9 Logik der Datenbehandlung 88
6.3 Funktionalität 89
6.3.1 Brückenadministration 89
6.3.2 Inspektionsübersicht 90
6.3.3 Hinzufügen von Inspektionen 91
6.3.4 Markieren von Schäden 93
6.4 Abgleich Anforderungsprofil und Funktionspalette 94
7 User Tests 96
7.1 Ablauf und Konzept 96
7.2 Durchführung und Ergebnisse 97
7.2.1 Interpretation der Ergebnisse der aufgabenbasierten Tests 98
7.2.2 Interpretation der Ergebnisse aus den spezifischen Fragen 99
7.2.3 Mögliche Limitationen 101
8 Fazit 102
Literaturverzeichnis 104
Abbildungsverzeichnis 110
Tabellenverzeichnis 112
Listingverzeichnis 113
Anhang 114
Experten und Expertinnen Fragen und Testaufgaben 114

1 Einleitung

Diese Arbeit befasst sich mit dem Thema Entwicklungsoptionen zur Unterstützung von Brückeninspektionen mit mobilen Geräten. Die Hauptcharakteristiken liegen im Bereich der Optimierung eines Prozesses und aktuellen Technologien in der App Entwicklung. Das Thema der Prozessoptimierung ist die Brückeninspektion. Um den Prozess zu analysieren, wird dieser in Teilbereiche aufgeteilt. Zusätzlich finden Interviews mit Expertinnen und Experten sowie einer Institutionsperson im Bereich Bahntechnologie & Mobilität statt. Durch diesen Input wird dann ein Anforderungsprofil erstellt, das Profil, der Grundbaustein und das Grundkonzept. Außerdem wird eine Literaturrecherche durchgeführt, die aktuelle Forschungsprojekte im Bereich der Inspektion von Bauwerken und der Brückeninspektion analysiert. Der nächste Themenschwerpunkt ist die Technologie, auf dem Konzept aufbauend, wird eine weitere Literaturrecherche durchgeführt. Schwerpunkt der State of the Art Analyse sind aktuelle Entwicklungsoptionen zu verwandten Themen wie die native, hybride oder webbasierte App Entwicklung. Auf der Basis des Anforderungsprofils und der Entwicklungsanalyse wird eine Applikation entwickelt. Die App soll den Experteninnen und Experten den Ablauf einer Brückeninspektion erleichterten und den Verwaltungsaufwand minimieren. Durch eine gute Handhabung soll der Ablauf einer Inspektion Zeit und Ressourcen sparen. Eine Auswertung der User Tests und das Fazit bilden den Abschluss der Forschung.

1.1 Problemstellung und Relevanz

Die Branche Brücken und Tunnelbau hatte einen im Jahr 2014 einen Umsatz von 1.806 Milliarden Euro und die Prognose der Statista sagt voraus, dass der Umsatz bis zum Jahr 2020 auf bis 2.432 Milliarden Euro steigen wird (Statista, 2017a). Diese Zahlen erläutern, dass in dieser Branche in den nächsten Jahren mit weiterem Wachstum zu rechnen und Potential vorhanden ist. Dadurch wird auch die Inspektionsarbeit von Brücken im Fokus bleiben. Schäden bei Brücken sind schwere Sicherheitsmängel und daher werden in diesem Bereich bei den Bauobjekten laufend Inspektionen durchgeführt. Das Intervall zwischen den Überprüfungen ist unterschiedlich und hängt vom jeweiligen Bauobjekt ab.

Ein weiterer wichtiger Punkt ist, dass die Brückeninspektionen nur von Fachpersonal durchgeführt werden dürfen. Das Fachpersonal, das die Inspektionen vor Ort ausführt, muss außer einfachen statischen Kenntnissen vor allem praktische Erfahrung im Massivbrückenbau aufweisen. Mitarbeiterinnen und Mitarbeiter, die als Ausführende (PolierIn, BauleiterIn, KonstrukteurIn, SpannmeisterIn etc.) oder als Bauüberwachungsorgane entsprechende Erfahrungen, insbesondere auch bei Spannbetonbrücken sammeln konnten, werden diesen Anspruch am ehesten erfüllen (Österreichische Bundesbahnen, 1996).

Das Fachpersonal dokumentiert die Brückeninspektion handschriftlich und legt diese Dokumente dann im Archiv ab. Die Inspektionsdaten werden nicht digital erfasst, d.h. die Daten sind in keiner Datenbank vorhanden. Im Zeitalter von Smartphones, Tablets, Cloud Computing und High Speed Internetleitungen ist das wahrscheinlich langfristig gesehen, keine Variante für die Zukunft. Eine weitere negative Eigenschaft bei der derzeitigen Variante ist die mögliche Fehlerhaftigkeit und auch Ungenauigkeit. Durch das handschriftliche Dokumentieren besteht ein größerer Risikofaktor, da möglicherweise die Schrift einer Inspektorin oder eines Inspektors nicht gelesen werden kann. Weitere Fehlerquellen können bei der Dokumentenarchivierung entstehen, denn bei der Ablage von Dokumenten in Papierform können Fehler passieren oder auch, dass nach Einsicht in ein Dokument einfach ein Datenblatt nicht ordnungsgemäß zurückgelegt oder vergessen wird. Eine weitere relevante Problemstellung ist die Flexibilität, denn durch die Papierablage gibt es kein flexibles Verwalten von Dokumenten. Es kann nicht von mehreren Standorten und mehreren Personen auf ein Dokument sofort zugegriffen werden.

Wenn man das derzeitige Verfahren bzw. System digitalisieren möchte, muss die Bewertung, Wichtigkeit und die Eigenschaften einzelner Prozesse sorgfältig analysiert werden. Als Arbeitsmittel können Informationstechnologien, die Art der Arbeitsausführung und die Gestaltung der Kontextbedingungen strukturieren, insofern also die Arbeitspraxis formalisieren oder individuelle Handlungsspielräume eröffnen und dadurch die Arbeitspraxis informalisieren (Funken & Schulz-Schaeffer, 2008).

1.2 Ziele der Arbeit

Ziel der Arbeit ist es, den Brückeninspektionsprozess durch den Einsatz von Entwicklungstechnologien zu unterstützen, um mehr Flexibilität und Transparenz bei der Archivierung für das zuständige Fachpersonal zu bieten. Eine Analyse des derzeitigen Prozesses und daraus ein Anforderungsprofil zu erstellen. Gleichzeitig soll auch ein System entwickelt werden, um die Daten zu verwalten, zu

organisieren und zu optimieren. Die Handhabung soll schnörkellos, übersichtlich und leicht bedienbar sein.

Durch den Einsatz von mobilen Technologien und einem skalierbaren System sollen mögliche Fehlerquellen minimiert werden, damit keine Folgeprobleme entstehen können. Ein Beispiel für ein Folgeproblem wäre, wenn Risse, Abbröckelungen oder Korrosionen unleserlich oder mangelhaft dokumentiert wurden und dadurch Sicherheitsprobleme für die Brücke entstehen.

Nicht-Ziele sind es, Struktur oder Richtlinien der Inspektion zu verändern, da die Richtlinien von Experteninnen und Experten im Bereich des Brückenbaus entworfen wurden. Ein weiteres Nicht-Ziel ist es, eine Applikation zu entwickeln, die Performanceanalysen der Brücke erstellt. Das heißt es sollen nicht Risse, Abbröckelungen oder Korrosionen automatisch erkannt und diese nach Stärke der Degradationen bewertet oder Klassifizierungen erstellt werden. Außerdem soll es keine Plattform werden, mit einem allgemein definierten Pattern, die entwickelnde App soll ausschließlich für die Brückeninspektion angedacht sein.

1.3 Forschungsfragen und Hypothesen

Aus der Problemstellung und den Zielen ergeben sich folgende Forschungsfragen:

F1: Wie kann eine Prozessoptimierung bei einer Brückeninspektion aussehen und wie können mobile Anwendungen in der Entwicklung in diesem Themengebiet eingesetzt werden?

F2: Wie sieht ein konkreter Prozess für eine Brückeninspektion aus?

F3: Wie empfinden Experteninnen und Experten die Handhabung und den Umgang mit der Applikation?

F4: Welche Anforderungsoptionen und Entwicklungsoptionen müssen für eine solche Applikation für die Umsetzung gegeben sein?

F5: Welche Vorrausetzungen müssen erfüllt werden, um dem Anforderungsprofil und der Prozessanalyse gerecht zu werden?

Es wird erwartet, dass sich der Vorgang einer Brückeninspektion durch die Unterstützung einer Anwendung verbessert. Im Bereich der Schadensklassifizierung bei einer Inspektion werden Zeitoptimierungen erwartet. Voraussetzung einer Verbesserung in den diversen Bereichen ist eine korrekte und saubere Handhabung. Die Verbesserungsmöglichkeiten sind aber nur dann gegeben, wenn die Benutzerinnen und Benutzer der Applikation den Prozess auch ordnungsgemäß anwenden.

1.4 Methodik und Aufbau der Arbeit

Der erste Teil der Literaturrecherche ist zum aktuellen Stand im Forschungsbereich Inspektionen durch Unterstützung von mobilen Geräten durchgeführt. Es wurden Analysen und Untersuchungen durchgeführt, welche Technologie für die Umsetzung der Applikation verwendet wurde. Es wurden Studien zu analogen Prozessen mit mobilen Technologien untersucht. Die Studien wurden auch auf Mehrwerte und Optimierungen für Benutzerinnen und Benutzer analysiert. Die Basis für die Recherchearbeit sind Forschungsprojekte und Studien, die in wissenschaftlichen Datenbanken wie IEEE[1] und ACM[2]. Außerdem findet eine Konkurrenzanalyse zu einem Brückeninspektionstool statt.

Der zweite Teil der Literaturrecherche ist die State of the Art Analyse im Bereich der mobilen Entwicklungsoptionen. Hier werden verschiedene Methoden und Ansätze verglichen. Der Schwerpunkt dieser Analyse sind Web, Hybrid und Native Applikationen. Außerdem werden aktuelle Forschungsprojekte mit diesem Themenschwerpunkt verglichen.

Im Kapitel Problemanalyse wird der derzeitige Prozess der Brückeninspektion analysiert. Dazu sollte eine Methodik oder Konzept gefunden werden, um den Prozess und die Teilprozesse zu analysieren. Ziel ist es, die Stärken und Schwächen der Prozesse zu finden. Durch diese Analyse werden Anforderungen und Kriterien definiert. Außerdem werden vorhandene Dokumente, die im Bereich der Brückeninspektion vorkommen auch in das Anforderungsprofil eingearbeitet. Weitere Kriterien werden anlässlich eines Experteninterviews durch eine Expertin der FH St.Pölten im Bereich Bahntechnologie & Mobilität festgelegt. Durch diese Prozessanalyse sollten das Anforderungsprofil und die Richtlinien für die Entwicklung des Prototyps abgedeckt sein.

Im Kapitel Prototyp wird die Applikation bzw. der Prototyp erstellt. Es wird erklärt, wie die Umsetzung erfolgt ist. Welche Kriterien, Anforderung und Ansätze der Schwerpunkt in der Implementierung waren, sind im Kapitel Anforderungen und technische Entscheidungen definiert. Dazu wird beschrieben, welche Auswahlkriterien bei Wahl der mobilen Entwicklungstechnologie ausschlaggebend waren.

Im Kapitel User Tests geht es um die aufgabenbasierten User Tests. Dazu werden drei User Tests mit Expertinnen und Experten durchgeführt. Zusätzlich wird das Setup für die Testumgebung definiert und erklärt, wie die Dokumentation der User

[1] https://ieeexplore.ieee.org/
[2] https://dl.acm.org/

Tests abläuft. Ausgang der User Tests wird eine Beschreibung mit dem Umgang der Handhabung des Interface und der Funktionalitätspalette sein.

Es erfolgt dann eine Gliederung der Ergebnisse, in der auch Anmerkungen und Verbesserungen definiert werden. Außerdem werden mögliche Schwächen oder Stärken im Anforderungsprofil erkannt. Durch diese Evaluierungen kann dann im Kapitel Fazit eine Schlussfolgerung erstellt werden.

1.5 Definitionen

In diesem Abschnitt der Arbeit werden die Begriffe, Fachwörter und Definitionen erklärt, die in der Arbeit verwendet werden.

1.5.1 Mobile Entwicklungstechnologien

Unter diesem Begriff versteht man den Einsatz von Technologien wie Web, Native oder Hybrid basierte Technologien. Diese Programmierungsansätze sind für die Umsetzung von Applikationen im mobilen Bereich gedacht. Darunter fallen Tablets, Smartphones oder auch Smartwatches.

1.5.2 Container

Die Bezeichnung Container ist ein allgemeiner Begriff, der eine Verschachtelung repräsentiert. Als übergeordnetes Element das andere Elemente noch beinhaltet. Die Bezeichnung ist oft mit mobilen Entwicklungsmethoden verankert, speziell im Bereich der hybriden Technologien. Hybride Container sind Verschachtelungen die in einer Web Applikationen laufen. Auch als Container werden übergeordnete Elemente in einer HTML Struktur bezeichnet.

1.5.3 MVC

MVC (Model View Controller) ist ein Designpattern für die Entwicklung von Software und besteht aus drei Komponenten. Die Model Komponente ist verantwortlich für die Struktur der Daten. Die View Komponente stellt die Benutzeroberfläche dar. Die Komponente Controller ist verantwortlich für die Steuerung und ist die Schnittstelle zwischen Model und View. Benutzeraktionen werden vom Controller aufgefasst und dementsprechend verarbeitet.

1.5.4 Routing

Der Begriff Routing ist die Interpretation das Inhalte per URL angezeigt werden. Sprich es ist eine Abhängigkeit zwischen Inhalt und URL vorhanden. Der Aufruf von URLs dient auch als Hilfsmittel. Durch den Aufruf werden Aktionen klarer definiert und für die Benutzerin oder den Benutzer verständlicher beschrieben.

Beispiele für Routing ist das interagieren durch klickbare URLS in einer Navigationsleiste einer Webseite.

1.5.5 API Schnittstelle

Die Kennzeichnung API steht für Application Programming Interface und ist das Verbindungstück zwischen Backend und Frontend. In modernen Web Applikationen ist es ein verbreitetes Design-Pattern und ermöglicht eine Unabhängigkeit zwischen Frontend und Backend Entwicklerinnen oder Entwicklern.

2 Theoretische Ausgangslage

Dieses Kapitel befasst sich mit State of the Art Analysen im Bereich der Brückeninspektion und Prozessoptimierung. Ein weiterer Themenschwerpunkt ist die Beschreibung eines Konkurrenzproduktes. Der erste Themenbereich heißt Forschungsprojekte: hier soll ein Überblick über aktuelle Forschungsprojekte im Bereich der Brückeninspektion gegeben werden. Zusätzlich werden auch Analysen im Bereich der Prozessoptimierung durch mobile Technologien gemacht. Die Forschungsprojekte werden nicht nur das Thema Brückeninspektion behandeln. Der Grund dafür ist, dass ein anderer Blickwinkel auf das Thema projetziert wird. Welche Ergebnisse und Aussagen diese Analysen geliefert haben, wird in den Erkenntnissen der Forschungsprojekte erläutert.

Der zweite Themenschwerpunkt ist die Beschreibung einer Konkurrenzanalyse einer Brückeninspektionssoftware. Hier werden die Eigenschaften und die Funktionspalette der Applikation beschrieben.

2.1 Forschungsprojekte

2.1.1 Research and Application of Bridge Inspection and Maintenance Management System

Das erste Forschungsprojekt wurde von zwei chinesischen technischen Universitäten und einer Partnerfirma designt und implementiert. Das Thema des Projektes behandelt ein Wartungssystem und Inspektionsinstrument für Brückeninspektionen (Zhang, Liang, Chen, & Li, 2016). Die größte Motivation der Mitarbeiterinnen und Mitarbeiter an diesem Forschungsprojekt war, die wirtschaftliche und infrastrukturelle Sicht im Bereich der Brückeninspektion zu optimieren. In China gibt es ein gesundes Wachstum sowohl im Bereich der Wirtschaft, als auch im Bereich der Infrastruktur. Durch die Belastungen der Brücken wurden in den letzten Jahren immer mehr Schäden und Korrosionen festgestellt. Nach den Inspektionen wurden die anfälligen Bauarbeiten durchgeführt und es gab dadurch Probleme in der Infrastruktur. Ihr Standort für dieses Tool war die Stadt Kunming[3] in der Volksrepublik China. Nach den Ausbauarbeiten der Infrastruktur der Stadt bestand die Notwenigkeit, ein System zu integrieren, das ein Inspektionstool und ein Wartungssystem besitzt. So konnten die Bauarbeiten besser koordiniert und optimiert werden. Die Entwicklung

[3] https://de.wikipedia.org/wiki/Kunming

erfolgte in Kooperation mit dem Unternehmen Kunming Bridge and Tunnel Management Company Limited und mit der Universität School of Information, Yunnan University (Zhang u. a., 2016).

Die angewendeten Entwicklungsoptionen für das Projekt waren eine MySQL Datenbank, für die Frontendausgabe wurde Framework ASP.NET[4] von Microsoft verwendet. Das Framework ist ein serverseitiges Framework, das für Webseiten und Web Applikationen designt wurde. Entwickelt wurde mit C#, HTML und JavaScript im Microsoft Tool Visual Studio. Außerdem wurde die JavaScript Bibliothek ArcGis verwendet, das ein WebGIS System unterstützt. Unter dem Begriff WebGIS versteht man eine Applikation, deren Hauptfunktionalität auf Geodaten spezialisierte Webservices bzw. auf Geodienste zurückgreift. Um Datenkonsistenz und Sicherheit zu garantieren, wurde die Serverumgebung bei Kunming Bridge and Tunnel Management Company Information Center aufgesetzt.

Ihr Management Instrument wurde in mehrere Bereiche eingeteilt und deckt ein großes Spektrum an Funktionen ab. In der Abbildung 1 wird erklärt, welche Funktionalität und Module das Tool besitzt.

[4] https://www.asp.net/

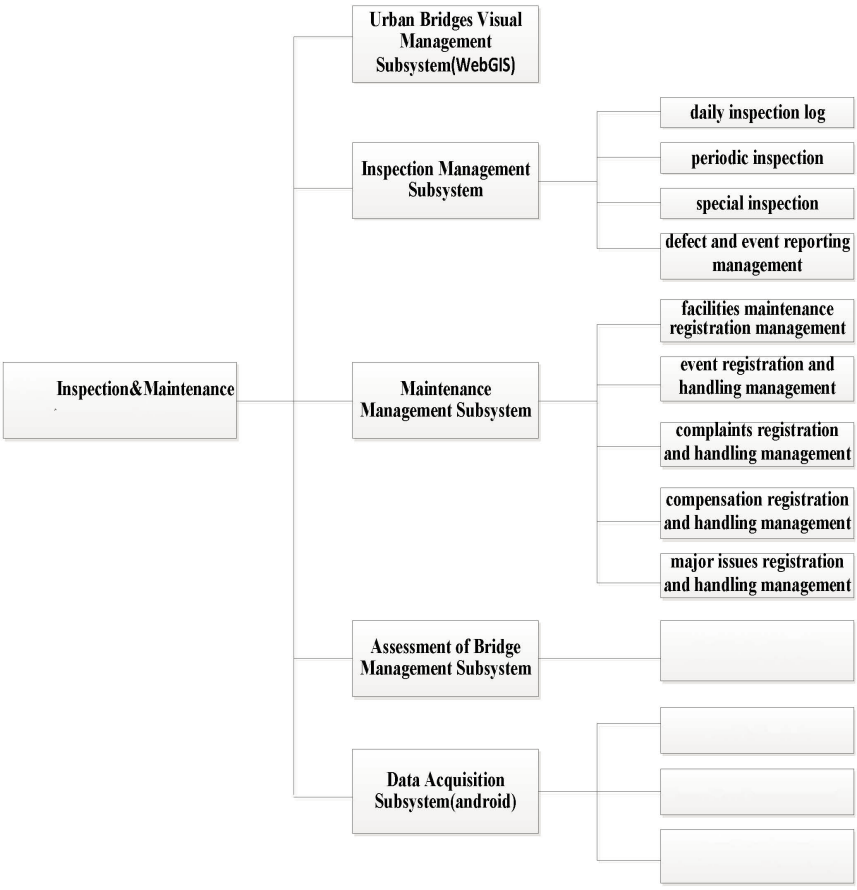


Abbildung 1: System Module der Applikation Inspection Management Subsystem und Maintenance Management Subsystem (Zhang u. a., 2016)

Die Analyse bezieht sich auf die beiden Hauptpunkte wie **Inspection Management Subsystem** und **Maintenance Management Subsystem**. Das entwickelte Inspektionsinstrument basiert auf tägliche, periodische und spezielle Inspektionen und zeichnet diese auf. Das Design Pattern basiert auf einem Brückeninspektionsstandard (Zhang u. a., 2016). Durch den Standard soll die Applikation dann für andere Städte auch nutzbar werden. Wie bei der österreichischen Variante der Brückeninspektion, wird auch hier auf Papierdokumente gesetzt. Zusätzlich zur Papiervariante wurde von den Entwicklerinnen und Entwicklern eine Datenbank für die Inspektionsaufzeichnung integriert. Durch diese Integration erwarten sich die Mitarbeiterinnen und Mitarbeiter eine Prozessoptimierung. Als Effekt soll dann die Koordination von Wartungsarbeiten ohne Verzögerungen funktionieren. Außerdem wurde in das Inspektionstool eine Fehler- und Event Reporting Funktion eingebaut. In Abbildung 2 wird dieses Modul als Ablaufdiagramm dargestellt und beschrieben.

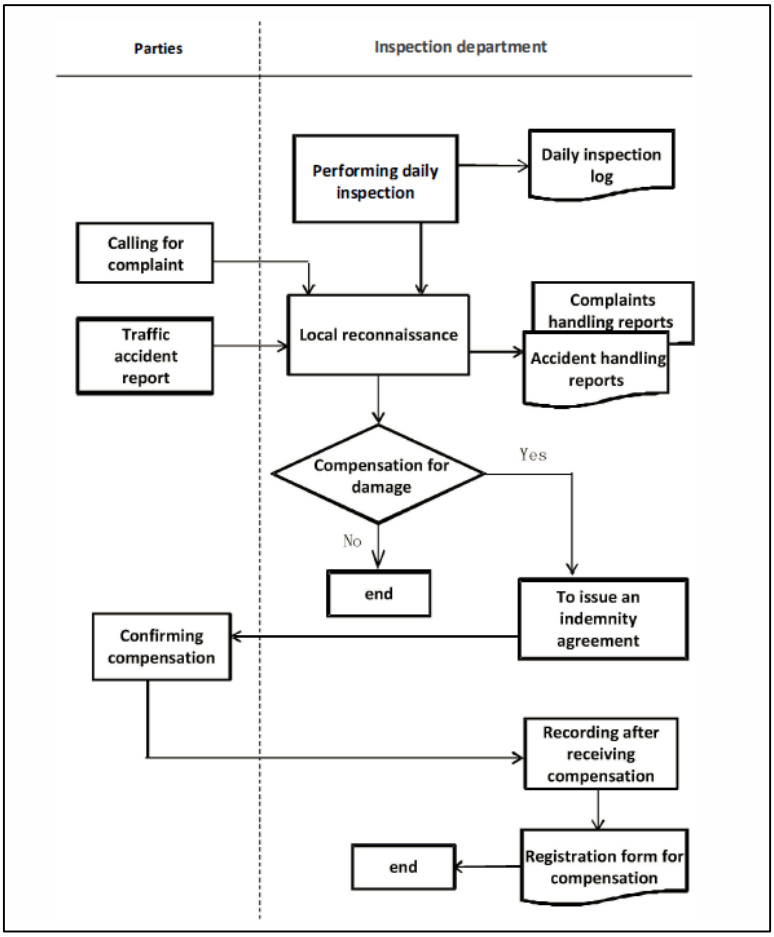


Abbildung 2: Ablaufdiagramm des Reporting Moduls der Applikation Inspection Management Subsystem und Maintenance Management Subsystem (Zhang u. a., 2016)

Das Reporting Modul ist in Intern und Extern strukturiert. Externe Parteien haben die Möglichkeit, Beschwerden einzureichen und zusätzlich können Umfälle und andere Ursachen gemeldet werden. Auch interne Mitarbeiterinnen und Mitarbeiter können Reporting Berichte erstellen. Ursachenaufklärung passiert in der internen Abteilung, dazu werden die Problemberichte in Kategorien eingeteilt und nach den Ursachen erforscht. Wenn tatsächlich Schäden für die externen Parteien entstanden sind, werden von der internen Fachabteilung Rückerstattungen ausgesendet. Diese müssen dann von der externen Partei bestätigt werden.

Der nächste wichtige Hauptpunkt ist das Managementsystem. Dieses Tool wurde hauptsächlich für das Personal entwickelt. Dadurch wird eine wirtschaftliche Kostenreduzierung in Form von Zeitersparnis erwartet. Durch eine saubere und effiziente Koordinierung des Personals soll die Optimierung stattfinden. Der Prozess ist so definiert, dass für das Wartungspersonal sogenannte Events zugeteilt und diese durch die diversen Abteilungen abgearbeitet werden. Die Abbildung 3 soll einen Einblick geben, wie die Prozessaufteilung funktioniert.

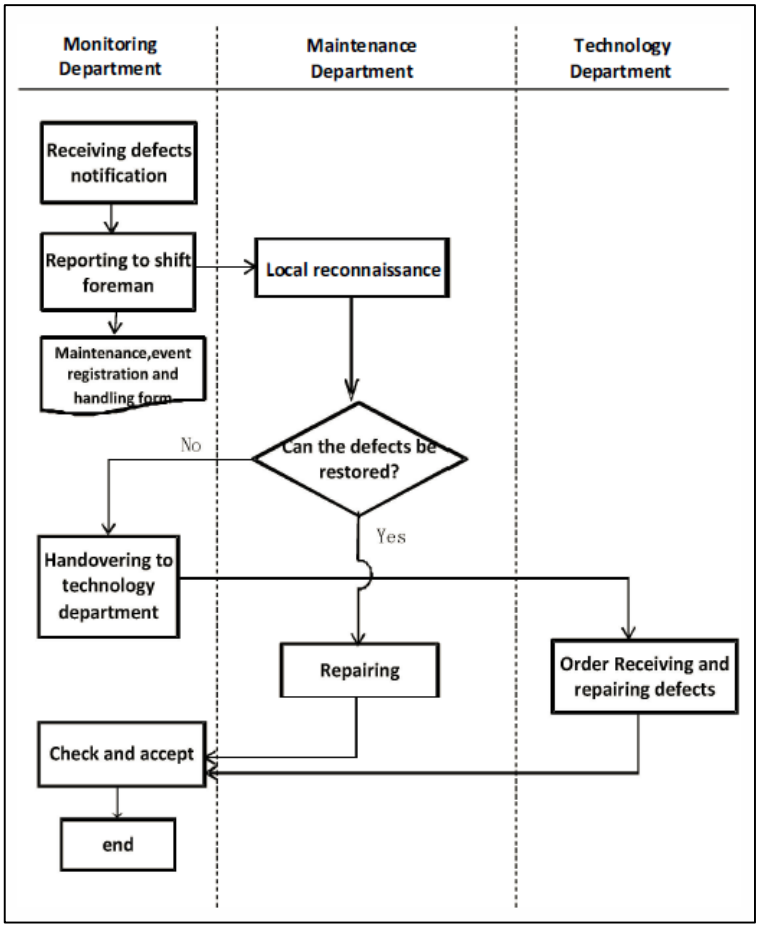


Abbildung 3: Ablaufdiagramm des Wartungssystems der Applikation Inspection Management Subsystem und Maintenance Management Subsystem (Zhang u. a., 2016)

Die Kontrollabteilung überwacht die Systeme und weist die Aufträge dem Wartungspersonal zu. Das Wartungspersonal überprüft den Auftrag und probiert das Problem zu beheben. Sowohl im Erfolgsfall, als auch im Problemfall wird ein Status an die Kontrollabteilung gesendet. Die Kontrollabteilung sendet einen weiteren Status an die Technologie Abteilung, falls das Problem nicht gelöst wurde. Für das Wartungspersonal wurde zusätzlich eine native Android Applikation entwickelt, die den Mitarbeiterinnen und Mitarbeitern ermöglicht, Daten zu erfassen. Die App hat zwei Kernpunkte: der erste Kernpunkt ist die Dokumentation von Schäden. Die Mitarbeiter können die Schäden mittels App dokumentieren. Der zweite Kernpunkt ist der Informationsgehalt. Die Applikation liefert Daten zum Bauobjekt, zu vorigen Inspektionen und sonstigen relevanten Daten für das Wartungspersonal.

Das Forschungsprojekt zeigt, wie die Implementierung und Umsetzung einer Prozessverbesserung im Bereich von Wartungsarbeiten und Brückeninspektionen aussehen kann. Die erste Version des Forschungsprojekts wird bereits von der Kunming Bridge and Tunnel Management Company Limited verwendet.

2.1.2 An effective practices, strategies and technologies in the service industry to increase customer loyalty using map indicator

Das folgende Forschungsprojekt behandelt den Optimierungsprozess eines Customer-Relationship-Management (CRM) Systems. *Was genau ist ein CRM System?* CRM ist als ein strategischer Ansatz zu verstehen, der zur vollständigen Planung, Steuerung und Durchführung aller interaktiven Prozesse mit dem Kunden genutzt wird. CRM umfasst das gesamte Unternehmen und den gesamten Kundenlebenszyklus und beinhaltet das Database Marketing und entsprechende CRM-Software als Steuerungsinstrument („Definition » Customer Relationship Management (CRM) « | Gabler Wirtschaftslexikon“, o. J.).

Das System wurde entwickelt, damit Kleinunternehmen mit ihren Kunden interagieren können. Eine Applikation für ein CRM System ist keine Neuheit, Ziel des Projekts war es, eine der Hauptkomponenten eines CRM Systems, zu verbessern. Um die Komponenten zu finden, die den Fokus eines effizienten CRM Systems deklariert, wurde eine Literaturrecherche gemacht. Nach dieser Analyse wurden Features definiert und dadurch das Anforderungsprofil erstellt. Um die Funktionen zu gliedern, wurde in Kunden, Unternehmen und Administrator aufgeteilt. Zusätzlich war es ein Ziel, die Kundenloyalität zu steigern. Um eine genaue Auswertung zu bekommen und dann Analysen zu erstellen, wurden Tracking Indikatoren implementiert. Diese Indikatoren haben die Möglichkeit, jeden Status und Interaktion der Userin oder des Users aufzuzeichnen. Durch diese Funktionalität soll die Loyalität der Kunden gesteigert werden. Die Definition eines loyalen Kunden laut ihres Forschungsprojekts sind Benutzerinnen und Benutzer die oft mit der App handhaben auch die loyalsten Kunden (Geetha, Safa, Saranya, & Subburaj, 2017).

Die Zielplattform für die Entwicklungsoptionen war eine native Android App. Es wurde aber hauptsächlich die Android Web View benutzt und die Datenverarbeitung lief auf einem zentralen System. Das Backend System wurde mit PHP programmiert und die Datenbank im Hintergrund war eine rationale Datenbank(MySQL). Der nächste Hauptpunkt war ein sauberes Interface Design, da ihre Zielgruppe mobile Userinnen und User sind. Das Interface Design wurde basierend auf den 8 goldenen Regeln von Ben Shneiderman entworfen (Shneiderman, 1987).

2.2 Erkenntnisse der Forschungsprojekte

Nach der Literaturrecherche kann man zur Erkenntnis kommen, dass das Thema Brückeninspektion durch mobile Entwicklungstechnologien ein großes unbetretenes Gebiet ist. Ein möglicher Hauptgrund könnte sein, dass

Entwicklerinnen und Entwickler solcher Applikation ihre Arbeiten, Dokumentationen und Forschungsarbeiten nicht freigeben möchten und noch in Entwicklung sind. Eine weitere Erkenntnis ist, dass fast immer nach denselben Schemata vorgegangen wird. Der Prozess wird definiert und es werden Stärken und Schwächen analysiert und Kriterien definiert, diese sind dann das Anforderungsprofil für die Entwicklung. Bei der Entwicklung war das Ergebnis breiter gefächert, es sind sowohl Webtechnologien als auch die native Entwicklungstechnologie zum Einsatz gekommen. Ausschlaggebend ist das Anforderungsprofil der Optimierung.

2.3 Konkurrenzanalyse: SIB-BAUWERKE

Das Programmsystem SIB-BAUWERKE ist eine Entwicklung der Straßenbauverwaltungen von Bund und Ländern in Deutschland, die Eigentümer dieses IT-Produktes sind das Unternehmen WPM Ingenieure. („WPM Ingenieure GmbH“, 2018). Dieses Tool wurde entwickelt für das deutsche Bundesministerium für Verkehr, Bau und Stadtentwicklung und ist ein Programm für die Verwaltung von Bauwerksdaten. Durch die Applikation besteht die Möglichkeit, Bestandsmaterial wie Bilder und Pläne einzubinden. Eine zusätzliche Eigenschaft des Tools ist es, bei Straßenbauobjekten Inspektionen durchzuführen.

Das heißt, es besteht auch die Möglichkeit mit der Applikation Brückeninspektionen durchzuführen. Die App ist eine Desktop-basierte Anwendung. Die Inspektorinnen und Inspektoren nehmen als Hilfsmittel ein Notebook zur Begutachtung mit. Schäden und Mängel werden nach den Standards und Normen des Bundesministeriums bewertet und es gibt einen Katalog, wo der jeweilige Schaden oder Mangel ausgewählt werden kann. Aus den einzelnen Schadensbewertungen wird dann am Ende durch das Programm automatisch eine Zustandsnote generiert. Sie beschreibt dann die Funktionalität und Tauglichkeit des Bauobjektes.

Um die Verwaltung für das Fachpersonal überschaubarer zu gestalten, wurde die Ansicht *Übersichtsblatt* erstellt. Das Übersichtsblatt fasst die wichtigsten Daten eines Teilbauwerks zusammen und stellt sie in übersichtlicher Form dar. Für die möglichen Bauwerksarten Brücke, Verkehrszeichenbrücke, Tunnel, Lärmschutzwand, Stützbauwerk und sonstiges Bauwerk unterscheidet sich das Übersichtsblatt geringfügig durch die Spezifikation der Bauwerksdaten(WPM Ingenieure GmbH, 2013).

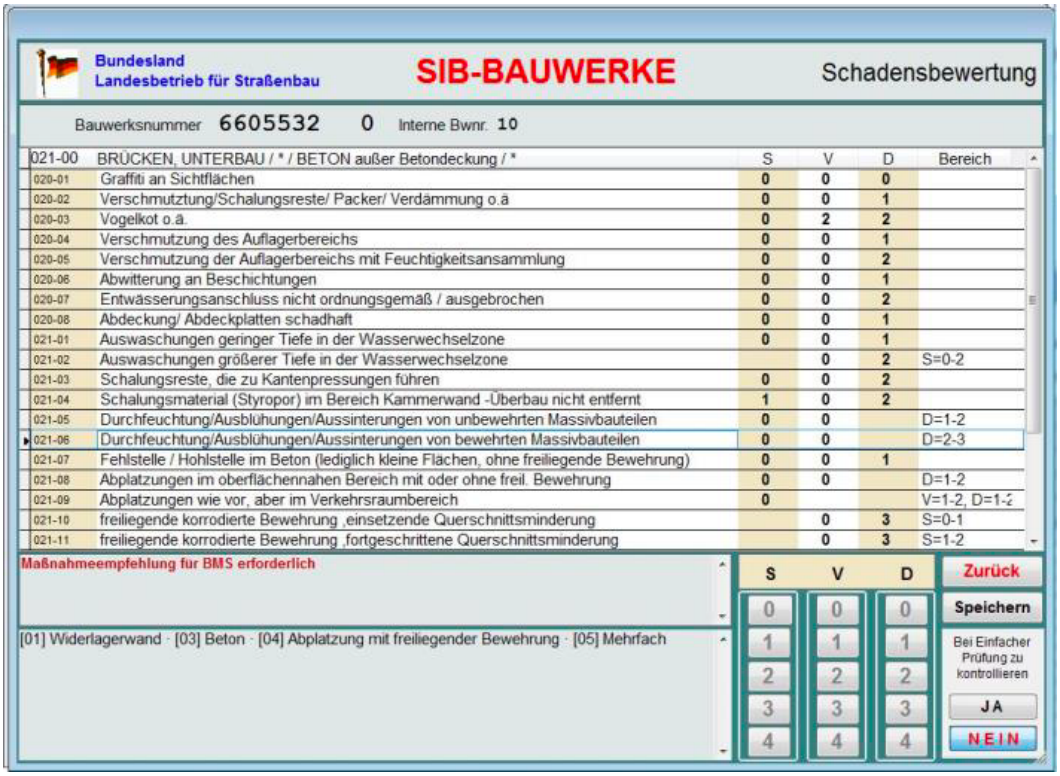


Abbildung 4: Anwendungsscreen von SIB-Bauwerke bei Erstellung von Einzelschäden

SIB-Bauwerke verwendet eine rationale Datenbank und die Kommunikation zwischen Datenbank und Client agiert mit einer Schnittstelle von Microsoft. Die rationale Datenbank läuft auf einem Microsoft SQL-Server.

3 Mobile Entwicklungstechnologien

Ein weiterer Punkt der Literaturrecherche ist die Analyse von mobilen Entwicklungsoptionen. Dazu wird untersucht, was der neuste Stand der Technik ist und welche Technologien im Einsatz sind. Es werden die Vorteile und Nachteile der Technologien erläutert und erklärt, welche Anwendungsgebiete abgedeckt werden.

Die mobilen Entwicklungsoptionen unterteilen sich in folgende drei Kategorien: native Entwicklungstechnologien, hybride Entwicklungstechnologien und webbasierte Entwicklungstechnologien. Der erste Teil der State of the Art Analyse beginnt mit den Webbasierten Entwicklungstechnologien. Außerdem wird der Ansatz der Progressive Web Apps (PWA) auch analysiert und untersucht. Die Ursache für diese Strukturierung in diesem Kapitel ist, dass Hybride Applikationen auf Web Technologien aufbauen. Dann wurden Ansätze und Methoden für hybride Entwicklungstechnologien untersucht. Weiters wurden die Eigenschaften von nativen Applikation an den großen zwei Marktführen Apples iOS[5] und Googles Android[6] untersucht.

Zusätzlich gibt es einen direkten Vergleich der Entwicklungstechnologien, in Form einer tabellarischen Darstellung.

3.1 Webbasierte Entwicklungstechnologien

Das Themengebiet webbasierte Entwicklungstechnologien differenziert sich von nativen und hybriden Technologien. Der Schwerpunkt dieser Entwicklungsform dreht sich um die Entwicklung von Webseiten oder den sogenannten Web Applikationen. Die Ausführung einer Web App findet immer im Browser über die URL Eingabe statt. Damit muss eine Internetverbindung gegeben sein, um die Applikation auszuführen und deshalb gibt es keine Distributionsplattform, wie den App Store oder Play Store. Mobile Geräte besitzen jedoch die Möglichkeit, mit einem Symbol oder sogenannten Lesezeichen, die Web Applikation am Homescreen zu verknüpfen. Das Ziel ist es, durch Web Technologien wie HTML5, JS und CSS für die Benutzerin oder den Benutzer ein natives Look & Feel zu

[5] https://www.apple.com/at/ios/ios-11/

[6] https://www.android.com/

erzeugen. Wie die Architektur einer Web App aufgebaut ist, wird in Abbildung 5 dargestellt.

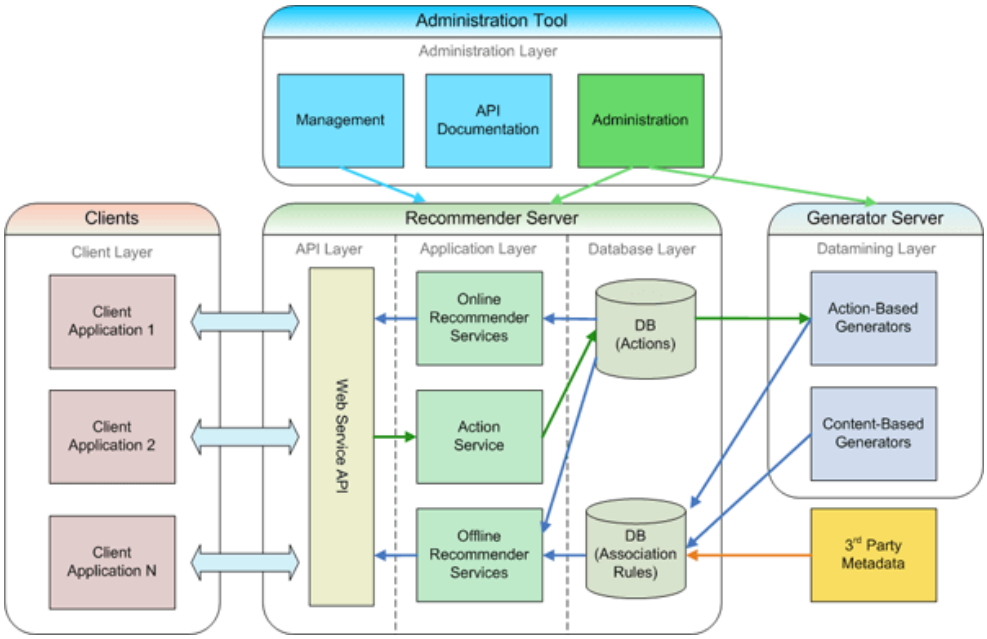


Abbildung 5: Architektur einer Web Applikation („Easyrec“, 2017)

Die Funktionalität ist auf den serverseitigen Schichten angesiedelt und organisiert den Zugriff auf die Datenbankschicht. Je nach Spezifikation wird die benötigte Datenbank installiert und konfiguriert. Um die Kommunikation zwischen Server und Client zu gewährleisten, gibt es den API Layer. Die API Schicht handhabt und organisiert die Verwaltung von Anfragen - in der Fachsprache auch als Requests bekannt. Der Client sendet einen Request an den Server, durch die API Schnittstelle wird der Request zugeordnet und führt die entsprechende Funktionalität im Applikation Layer aus. Nach Ausführung der Funktionalität wird die Antwort wieder durch die API Schicht an den Server gesendet. Spezifisch nach Anwendungsgebiet können dann Administrationsbereiche definiert werden, die direkt auf den Applikation Layer zugreifen.

Der Bereich auf den die Userin und der User auf der Client Seite zugreift, wird auch als Frontend bezeichnet. Heutzutage liefern einige Softwarebibliotheken, Lösungen für mobile Web Anwendungen wie JQuery Mobile, Sencha Touch, JQTouch, Ionic und weitere Frontend Technologien (Xanthopoulos & Xinogalos, 2013). Durch diese Frontend Frameworks sollen die Benutzerin und der Benutzer ein besseres Gefühl für Usability, Handhabung und Performance bekommen. Diese Frontend Technologien bauen meist auf JavaScript basiertem Code auf.

Web Apps werden durch den Browser ausgeführt und benötigen eine Internetverbindung. Genau diese Eigenschaft ermöglicht keine Offline

Funktionalität. Ein alternativer Löschungsansatz für diese Problemstellung wäre der Einsatz von bekannten Offline Datenbanken, wie CouchDB (Apache, 2017). Ein weiterer Ansatz wäre der Zugang von Web Storage, Indexed DB oder Cache der Applikation. Der Web Storage ist ein clientseitiger Speicher und ist unterteilt in Session Storage und Local Storage (Park, Kim, & Lee, 2016). Die Speicherkapazität des Web Storage ist um die 5 MB groß. Die Indexed DB ist wie CouchDB eine sogenannte NoSQL Datanbank. Die Indexed Storage Lösung hat den Vorteil, dass eine große Menge an Daten geliefert werden kann und es keine Limitationen gibt. Ein Nachteil für die Implementierung der Schnittstelle ist, dass eine gewisse technische Basis im Bereich der Programmierung für die Anwenderin bzw. den Anwender vorausgesetzt wird. Letzte Methode für Offline Funktionalität ist der Cache Speicher für Applikationen. Dieser wurde für Offline Anwendungen konzipiert und entwickelt. Der App Cache erlaubt Zugriffe auf die Web Apps, auch wenn das System offline ist.

Es gibt einige JavaScript Frameworks, die den Schwerpunkt auf mobile Web Applikationen gesetzt haben. Ionic, Sencha Touch, Meteor oder die Verwendung einer Frontend Bibliothek sind neue und praxisgängige Technologien im Bereich der mobilen Entwicklungstechnologien.

3.1.1 Ionic

Das Open Source Framework Ionic wurde von der Entwicklungsfirma Drifty Co entwickelt und ist speziell für mobile Apps entwickelt worden (Ionic, 2017). Ionic ist ein webbasiertes Framework, das mit der hybriden Technologie Cordova[7] verbunden werden kann. Genau dieser Lösungsansatz macht es schwierig das Framework in die Kategorie Hybrid oder Web zuzuordnen. Ionic bietet die Möglichkeit, eine Web Applikation mit nativer Handhabung und Usability in eine Hybrid App zu rendern. Seit 2014 wurden bereits 4 Millionen Applikationen, die auf dem Framework Ionic basieren, auf den Distributionsplattformen veröffentlicht (Lynch, 2017).

Die letzte Version von Ionic basiert auf Googles Frontend Framework Angular. Dadurch wird auch nach dem MVC Pattern wie bei Angular die Architekturform der App verwaltet. Durch die Verknüpfung mit dem Framework Cordova ist es möglich, die Web Applikation als native App zu rendern. Außerdem bietet die Verknüpfung die Erweiterung von Cordova Modulen, damit die Web Applikation auf native Features wie Bluetooth, Push Benachrichtigungen oder den Touch Fingerprint Sensor, zugreifen kann. Außerdem ist bei der Entwicklung ein großer Fokus auf das Interface Design gelegt worden. Viele UI Elemente wie Navigation, Buttons oder Textfelder sind bereits vordefiniert und können in den Code eingebunden

[7] https://cordova.apache.org/

werden. Ionic hat auf ihrer Plattform ein Tool, das den Ablauf der App Entwicklung verbessern soll, basierend auf Prototyp Entwicklung, Testabläufe, Produktionsprozess und Monitoring (Griffith, 2017).

Ionic ist in diesem Sinne kein klassisches Web Framework, es ist mehr ein Web Framework für hybride Technologien. Diese Form von Entwicklung bietet je nach Anwendungsgebiet einige Vor- und Nachteile. Durch Web Entwicklungstechnologien kann eine native Applikation erstellt werden. Durch den Ionic Market ist es möglich, Erweiterungen und Designs für die Applikation zu verwenden. Durch diese Erweiterungen und die Verschachtelung diverser Container kann der Entwicklungsprozess darunter leiden, da die Fehlersuche und das Debugging ein größerer Aufwand sind, als bei den klassischen Web Applikationen.

3.1.2 Meteor

Das nächste Entwicklungsinstrument ist wieder ein JavaScript basierendes Framework, das auf Node.js aufbaut. Das Framework wurde speziell für die Webentwicklung entworfen und liefert zusätzlich noch die Möglichkeit, Applikationen in einem Cordova Container als Hybrid App laufen zu lassen. Als Programmiersprache wird JavaScript verwendet. Ziel ist es, mit einer Codebasis eine Applikation für Web, iOS, Android und Desktop erstellen zu können. JavaScript kommt am Client und am Server zum Einsatz (Meteor Development Group, 2017). Es ist möglich, reaktionsschnelle Anwendungen zu erstellen, die auf allen Clients jederzeit den aktuellen Status der Datenbank in Echtzeit reflektieren (Zeller, 2016). Zusätzlich ist die NoSQL Datenbank MongoDB integriert. Meteor ist ein sogenanntes Fullstack Framework, das die komplette Funktionalität der Applikation im Backend- und Frontendbereich abbildet. In der Abbildung 6 sieht man eine triviale Darstellung einer Meteor Applikation.

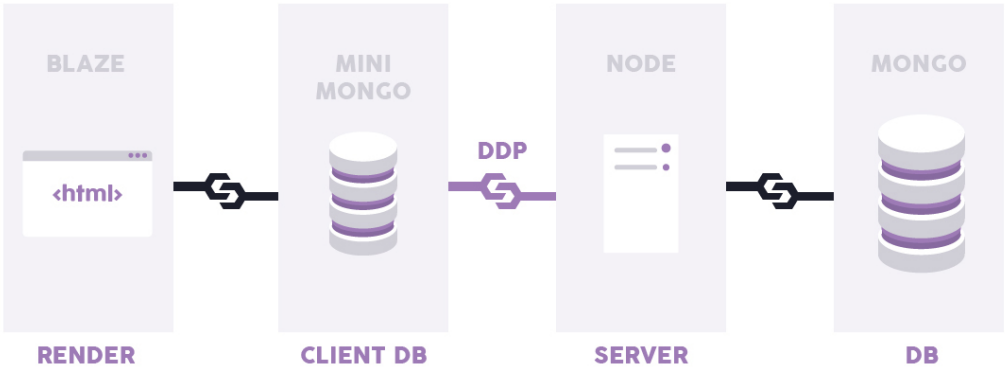


Abbildung 6: Ablaufdiagramm einer Meteor Applikation (Meteor Development Group, 2017)

Die Rendering Engine gestaltet den Prozess für die Darstellung und dazu wird das Modul Balze verwendet. Alternativ können andere Frontend Frameworks wie

Angular oder React als Client Abbildung bzw. als Ersatz von Balze verwendet werden. Auf der Client Seite gibt es eine Caching Ebene, in der Grafik wird es als Mini Mongo dargestellt. Das ist eine Kopie der Datenbank, die am Client liegt. Wenn Datensätze verändert werden, schickt der Client die Daten an den Server und Node verarbeitet dann die Daten in der MongoDB. Durch den Einsatz von Distributed Data Protocol (DDP) in Verbindung mit WebSockets wird es ermöglicht, Daten in Echtzeit abzurufen und zu verarbeiten. Im Code müssen nur auf Client und Server Seite jeweils ein Verbindungspunkt definiert werden - den Rest der Funktionalität bildet Meteor eigenständig ab (Vogelsteller, Strack, & Reyna, 2016).

Zusammenfassend bringt Meteor einige Vorteile: die Entwicklerin und der Entwickler können die komplette Applikation mit einer Programmiersprache umsetzen und durch die Konstellation mit dem Distributed Data Protocol kann Meteor Daten in Echtzeit verarbeiten. Dadurch können Funktionalitäten performanter integriert und umgesetzt werden. Mit der Verbindung von Cordova können auch Hybrid Apps für iOS und Android erstellt werden. Nachteile von Meteor ist der mögliche Deployment Prozess - Applikationen laufen nur auf Node Servern, die in der Regel mehr Ressourcen benötigen, als der Standard Apache Server. Fullstack Frameworks sind auch nicht für alle Arten von Applikationen einsetzbar oder man läuft Gefahr, für spezielle Bedürfnisse nicht das bestmögliche Mittel zu verwenden. Würde man sich von einer Full-Stack-Lösung distanzieren wollen und gewisse Bereiche mit anderen Sprachen oder Methoden ersetzen, ist das mit solchen Systemen manchmal etwas aufwändiger – oder zumindest ungewohnt.

3.1.3 Sencha Touch/Ext JS

Wie auch Ionic und Meteor ist Sencha Touch ein JavaScript Framework, das speziell für den mobilen Markt entwickelt wurde. Sencha Touch ist ein Produkt von Sencha, welches als eine JavaScript Bibliothek aus der Zusammenführung von den Projekten Ext JS, jQTouch und Raphaël entstanden ist (Sencha Inc., 2015). Auf der Sencha Webseite wird das Produkt nur mehr als Ext JS angeführt. Ext JS und dann Sencha Touch waren die Vorreiter von JavaScript Frameworks, die speziell für mobile Endgeräte entworfen worden sind. Aber durch die Beeinflussung von größeren Unternehmen auf dem Markt von mobilen Technologien, wurden die Marktanteile kleiner und es kam zur Fusion von Ext JS und Sencha Touch. Die klassischen Webstandards wie HTML5, CSS und JavaScript werden für die Erstellung von Applikationen verwendet. Ein wichtiger Bestandteil von Sencha Touch/Ext JS ist die Verwaltung von UI Komponenten. Der Hersteller liefert eine große Auswahl von vordefinierten Komponenten, die in das User Interface adaptiert werden können. Die Module wurden auf diversen Plattformen getestet um den Entwicklerinnen und Entwicklern browserunabhängig Sicherheit im UI zu geben. Sencha hat für ihr Produkt eine große

Entwicklungsumgebung erstellt, mit der der Entwicklungsprozess optimiert werden kann. Es gibt Features wie User Interface Oberfläche, Testumgebung und Code Editoren.

Wie auch die anderen JS Frameworks bietet Sencha Touch/Ext JS, eine Adaption zur Cordova und PhoneGap, um hybride Applikationen zu erstellen. Ein Nachteil von Sencha Touch/Ext JS ist, dass es in den letzten Jahren etwas in Vergessenheit geraten ist. Die Community von Entwicklerinnen und Entwicklern ist nicht so groß wie bei den Konkurrenzprodukten. In der Analyse Software Google Trends wurde eine Auswertung über das Suchmaschinenverhalten von Userinnen und Usern zu den Suchbegriffen Sencha Touch, Ext JS und Ionic im Bereich Web-Design und –Entwicklung gemacht (Google, 2017c).

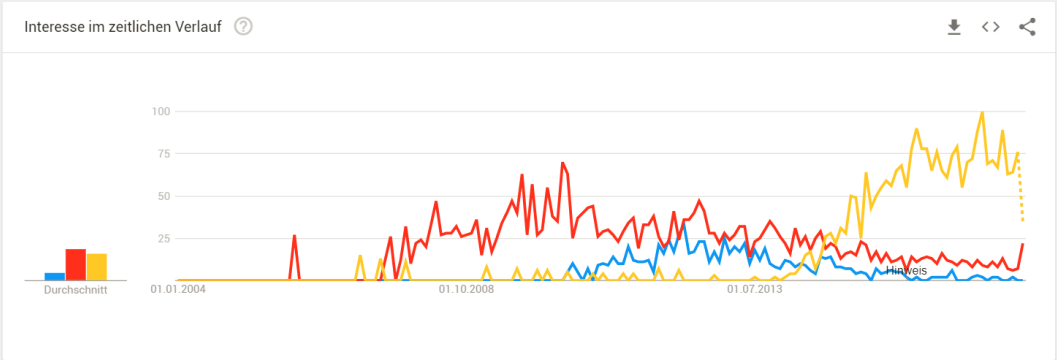


Abbildung 7: Auswertung von Sencha Touch/Ext JS durch Google Trends

Die rote Kurve ist Ext JS, blau zeigt Sencha Touch und gelb weist auf Ionic hin. Wie daraus ersichtlich, ist das Suchverhalten von Sencha Touch/Ext JS sehr rückläufig.

3.1.4 Frontend JS Bibliotheken

Klassische Web Technologien wie HTML und JavaScript bieten auch die Möglichkeit, Web Apps zu erstellen. Inzwischen gibt es einige Frontend JavaScript Bibliotheken, die für die Programmierung von Web Applikationen verwendet werden.

In dieser Analyse wurden folgende JS Frontend Bibliotheken herangezogen:

- Angular - alternative Bezeichung AngularJS
- React[8] - alternative Bezeichung ReactJS
- Vue[9] - alternative Bezeichung Vue.js

[8] https://reactjs.org/

[9] https://vuejs.org/

Die Auswahl dieser Bibliotheken wurde auf Basis der Beliebtheit in der sogenannten Gemeinschaft der Webentwicklerinnen und Webentwickler definiert. In der Tabelle 1 wurden die JS Bibliotheken anhand der Parameter der Open Source Plattform GitHub[10] verglichen.

Tabelle 1: JS Frontend Bibliotheken im Vergleich (Stand 10.01.2018)

Parameter	***Angular***	***React***	***Vue***
GitHub Sterne	*31.828*	*85.352*	*79.319*
GitHub Fork	*7.890*	*16.149*	*11.718*
Stack Overflow Fragen	*24.7768*	*67.772*	*12.620*

GitHub Sterne sind Parameter für die Bewertung des Projektes, es ist vergleichbar mit der Like Funktion in sozialen Netzwerken. Die Fork Funktion schafft die Möglichkeit vom aktuellen GitHub Projekt eine Kopie zu erstellen, an der die Webeentwicklerin oder der Webentwickler weiterarbeiten können. Stack Overflow ist die größte Plattform für Hilfestellung und Support im Bereich der Softwareentwicklung.

Wie anhand der Statistiken von GitHub zu erkennen ist, haben die Frontend Bibliotheken einen gewissen Stellenwert in der Szene der Webentwicklerinnen und Entwickler. Alle Projekte sind Open Source und liefern viele positive Eigenschaften, wie die performante Entwicklung von User Interfaces oder die Sicherheitsfunktionalitäten (Korotya, 2017).

Hinter Angular und React stehen Firmen wie Google und Facebook, die auch die wirtschaftlichen Mittel haben um das Framework richtig am Markt zu positionieren. Vue wurde von einem Programmierer entwickelt und durch die Entwickler Gemeinschaft hochgezogen.

Angular verwendet TypeScript, React und Vue verwenden den JavaScript Standard ES5 oder ES6. Jeder der JS Bibliotheken hat eine andere Systematik bei den sogenannten Core Features. Auch bei der Verwendung von Template Dateien gibt es Unterschiede, denn jedes der Frameworks hat verschiedene Template Rendering Systematiken (Neuhaus, 2017) .

Die Architekturformen haben auch gewisse Unterschiede - Angular zum Beispiel ist ein MVC Framework. React ist ein Framework, das nur die View behandelt,

[10] https://github.com

Controller und Model müssen selbst definiert werden. Vue hat das Architekturmuster Model View ViewModel(MVVM) und ist eine andere Form der MVC Architektur. Ziel eines MVVM ist, wie im MVC die Aufteilung der Schichten von Logik und Benutzerschnittstelle (Passaglia, 2017).

Ein weiterer wichtiger Bereich in der Webentwicklung ist die Performance der Bibliotheken. Um die Frameworks auf Performance zu testen wurde ein Benchmark Test herbeigezogen (Krause, 2015/2018). Vue hatte die besten Werte im Bereich der Zugriffe auf den Speicher. Das Ergebnis dieses Performance Tests ist in Abbildung 8 zu sehen.

Name	angular-v4.1.2-keyed	react-v15.5.4-redux-v3.6.0	vue-v2.3.3-keyed
ready memory Memory usage after page load.	4.8 ± 0.0 (1.3)	4.9 ± 0.1 (1.3)	3.8 ± 0.0 (1.0)
run memory Memory usage after adding 1000 rows.	10.9 ± 0.1 (1.4)	10.8 ± 0.1 (1.4)	7.5 ± 0.1 (1.0)

Abbildung 8: Vergleich der Performance bei Zugriff auf den Speicher (Neuhaus, 2017)

Der Testsieger im zweiten Benchmark Test ist ebenfalls Vue, wie aus Abbildung 9 ersichtlich ist. Bei genauerem Ansehen ist zu erkennen, dass in diesen Bereichen kaum drastische Performance Unterschiede ersichtlich sind. Dieser Benchmark Test hatte den Fokus auf die Verarbeitung von Daten.

Name	angular-v4.1.2-keyed	react-v15.5.4-redux-v3.6.0	vue-v2.3.3-keyed
create rows Duration for creating 1000 rows after the page loaded.	193.1 ± 7.9 (1.2)	212.2 ± 14.2 (1.3)	166.7 ± 8.6 (1.0)
replace all rows Duration for updating all 1000 rows of the table (with 5 warmup iterations).	197.4 ± 5.3 (1.2)	206.7 ± 7.3 (1.2)	168.5 ± 5.0 (1.0)
partial update Time to update the text of every 10th row (with 5 warmup iterations).	13.0 ± 4.5 (1.0)	18.0 ± 1.6 (1.1)	17.3 ± 2.9 (1.1)
select row Duration to highlight a row in response to a click on the row. (with 5 warmup iterations).	3.4 ± 2.3 (1.0)	8.7 ± 2.9 (1.0)	9.3 ± 1.7 (1.0)
swap rows Time to swap 2 rows on a 1K table. (with 5 warmup iterations).	13.4 ± 1.0 (1.0)	17.1 ± 1.3 (1.1)	18.3 ± 1.5 (1.1)
remove row Duration to remove a row. (with 5 warmup iterations).	46.1 ± 3.2 (1.0)	52.4 ± 1.7 (1.1)	52.6 ± 2.7 (1.1)
create many rows Duration to create 10,000 rows	1946.0 ± 41.8 (1.2)	1931.7 ± 35.6 (1.2)	1587.5 ± 33.9 (1.0)
append rows to large table Duration for adding 1000 rows on a table of 10,000 rows.	324.6 ± 10.1 (1.0)	366.4 ± 10.9 (1.1)	399.5 ± 11.0 (1.2)
clear rows Duration to clear the table filled with 10.000 rows.	379.9 ± 11.3 (1.5)	410.9 ± 9.8 (1.6)	254.5 ± 5.0 (1.0)
startup time Time for loading, parsing and starting up	84.3 ± 2.6 (1.5)	93.8 ± 6.9 (1.7)	56.6 ± 2.5 (1.0)
slowdown geometric mean	**1.14**	**1.23**	**1.06**

Abbildung 9: Benchmark Tests der JS Bibliotheken (Neuhaus, 2017)

3.1.5 Progressive Web Apps

Die Progressive Web Apps auch bekannt als PWAs ist der Kategorie der Web Applikationen zuzuordnen. Die Idee ist hinter Progressive Web Apps ist, dass die Distributionsplattform der Browser ist. Apps kommen zukünftig nicht mehr aus dem Store einer Plattform auf das Gerät, sondern über den Webbrowser der ohnehin auf jedem Mobilgerät installiert ist (Liebel, 2017). Um diese Idee im Segment der

Entwicklungstechnologien zu etablieren, sorgen Internetriesen wie Google. Durch Anleitungen und Support organisiert Google die Promotion von PWAs. Wie Web Applikationen werden auch PWAs durch den Browser aufgerufen, dabei wichtig ist, dass der Aufruf mittels HTTPS stattfindet. Mit dieser Eigenschaft sollen Sicherheitsfunktionen abgedeckt werden. Dann können die Userin und der User entscheiden, ob die Applikation installiert werden soll (Malavolta, 2016). Unter dem Begriff Installation im Bereich der Progressive Web Apps versteht man, die Applikation auf den sogenannten Homescreen zu verknüpfen. Die Applikation wird dann im Vollbildmodus wie eine native Applikation dargestellt und das Symbol am Homescreen aufgerufen. Durch die spezielle Caching Architektur wird die App schneller als normale Web Apps geladen.

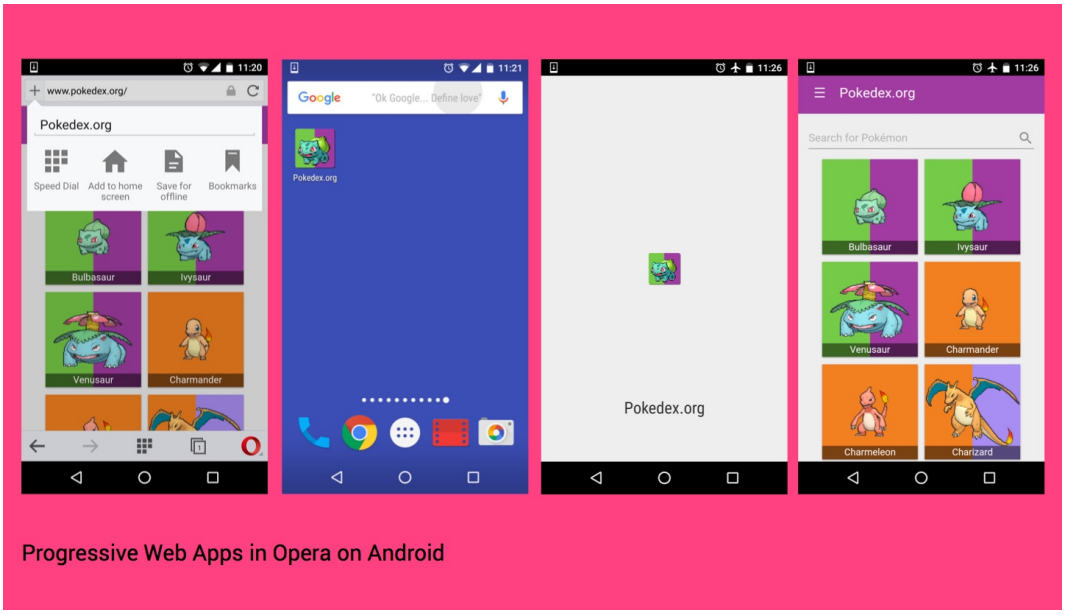


Abbildung 10: Darstellung der PWA von Pokedex.org (Lawson, 2015)

Eine weitere spezielle Charakteristik ist der Offline Modus. Dadurch ist es möglich, die Applikation zu verwenden wenn keine Internetverbindung vorhanden ist. Der Lösungsansatz um den Offline Modus darzustellen, ist der lokale Speicher des Browsers. Eine weitere Besonderheit ist die Möglichkeit, Benachrichtigungen zu versenden, selbst wenn die Applikation geschlossen ist. Außerdem sind Zugriffe auf Standort, wie bei normalen Web Apps gegeben. Die technische Entwicklung funktioniert mit dem sogenannten Service Worker. Die Service Worker sind für die Funktionalitätshandhabung mit Cordova für Hybrid Apps zu vergleichen.

Mit Service Worker lassen sich Abläufe separat von einer Website im Hintergrund durchführen (Möglichkeit für Aktualisierungen und Benachrichtigungen) oder Webanwendungen starten, während das Gerät offline ist (Gaunt, 2017). Wegen der Aktualität der Technologie unterstützen noch nicht alle Browser die

Funktionalität von Service Worker. Ein Beispiel dafür ist der Standardbrowser Safari von Apples iOS (Deveria, 2017). Bei der Entwicklung von progressiven Webanwendungen kann theoretisch jedes JavaScript Framework mit der Service Worker Bibliothek verknüpft werden. Google verweist mit Tutorials und Support auf ihr Frontend Framework Angular.

PWAs differenzieren sich im Bereich Aufbau der Architektur für die Caching Systematik von Standard Web Apps. Die Architektur wird in zwei Schichten aufgeteilt - in die Application Shell und in die Content Schicht. Beim ersten Aufruf der Application Shell wird die Schicht zwischengespeichert und wird dadurch Offline verfügbar. Das ermöglicht folgendes Feature: die App kann ohne Ladezeit aufgerufen werden. Die Content Schicht wird dann immer dynamisch mit aktuellen Inhalten nachgeladen. Ein Hauptmerkmal sollte sein, dass der Inhalt hauptsächlich serverseitig gerendert wird, um dadurch die Auslastung und Speicherkapazität auf der Client Seite zu reduzieren. Service Worker sollten als Modul für eine Verbesserung bzw. eine Erweiterung eingesetzt werden (Osmani, 2015).

In Bereichen wie Performance oder Hardware liefern progressive Web Applikationen signifikante Vorteile im Gegensatz zu den Standard Web Apps. Im Vergleich der Entwicklung werden mehr Ressourcen für die Integration der Service Worker benötigt, als bei normalen Web Applikation.

Da die Technologie noch nicht von allen Browsern unterstützt wird, besitzt der Ansatz von PWAs großes Potenzial und wird im mobilen Markt immer weiter vordringen. Bereits große Internetunternehmen wie Airbnb[11] haben Teile ihres Frontends als PWA ausgelagert (Neary, 2017). Nichtsdestotrotz haben PWAs nur Zugriff auf die sogenannten Low-Level Hardware Schnittstellen, wie Push Benachrichtigungen und Standort und können nicht dieselbe Funktionspalette wie hybride oder native Applikationen liefern. Einige Browser unterstützen bereits den Zugriff auf Kontakte, Kalender oder Batterie Sensor.

3.1.6 Funktionen und Fazit von Web Applikationen

Die Funktionsplatte von Web Applikationen ist im direkten Vergleich zu nativen Applikationen ziemlich eingeschränkt. Speziell der Zugriff auf die Hardware des Gerätes ist sehr limitiert. Die Hauptursache dafür ist der Zugang der Applikation - dieser erfolgt mittels Browser. Der Browser kann auf Funktionen wie die GPS Lokation oder die Kamera zugreifen. Durch PWA werden einige Funktionalitäten in die Web Applikation integriert, wie zum Beispiel Benachrichtigungen.

Die großen JavaScript Frameworks haben mittlerweile alle eine Anbindung zu nativen Containern durch Cordova. Durch diese Funktionalität kann die Hardware

[11] https://www.airbnb.at/

Limitation verbessert und zu einer hybriden Applikation generiert werden. Ein weiterer Blickpunkt ist das Anwendungsgebiet: mobile Applikationen bieten die Möglichkeit, mit einem einheitlichen Quellcode eine Anwendung für mehrere Geräte wie Smartphone, Tablet oder Desktop, zu erstellen. Dieser Aspekt ist der große Vorteil von mobilen Entwicklungstechnologien wie CSS, HTML und JavaScript. Ein weiterer Punkt ist, dass das Know-How für die Entwicklung von mobilen Web Apps nicht so spezifisch, wie von hybrid Apps oder nativen Apps, ist.

Der Einsatz von Web Technologien ist sehr an das Anwendungsgebiet angepasst. Bei der Entwicklung von Web Applikationen sollten Spezifikationen im Vorfeld abgebildet werden. Durch diese Abbildung sollte dann entschieden werden, ob die Funktionspalette von Web Apps das Anwendungsgebiet abgedeckt.

3.2 Hybride Entwicklungstechnologien

Was versteht man genau unter dem Begriff hybride Entwicklungstechnologien? Es gibt die native Appentwicklung und die Web Appentwicklung mit HTML, CSS und JavaScript. Hybride Anwendungen schließt die Lücke zwischen diesen beiden Optionen. Hybrid Apps besitzen teilweise native und webbasierte Komponenten. Die Distribution der Hybrid Applikationen laufen - wie bei den nativen Apps - über die Vertriebsplattformen der Betriebssystemhersteller. Eine Studie aus dem Jahr 2014 zeigt, dass von 1,100.000 Apps im Android Play Store über 129.000 Hybrid Apps waren; das war ein Marktanteil von 11.8% (Viennot, Garcia, & Nieh, 2014).

Hybride App werden mit sogenannten CPT (Cross Plattform Tools) entwickelt. (Ali & Mesbah, 2016). Die Funktionalität sieht so aus, dass die Entwicklerinnen und Entwickler Frontend Technologien wie HTML, CSS und JavaScript verwenden können, um die App zu programmieren. Dieser Code wird dann in einen Container mit einer Web Ansicht gepackt. Die Web Ansicht wird in einem nativen Container eingebettet. Der Quellcode der Applikation wird eigentlich vom Browser ausgeführt, der im Container läuft.

Durch diese nativen Eigenschaften kann dann die Installation der App auch über den App Store oder Play Store stattfinden. Für die Userin oder den User ist es oft kaum erkennbar, ob es sich um eine native oder hybride Applikation handelt. Die prominentesten Vertreter der CPT sind Adobe PhoneGap, Trigger.io, Appclerator Titanium, React Native und Adobe Air. Um näher auf die Marktanteile der CPT einzugehen, wird auf die Studie von Mohamed Ali und Ali Mesbah verwiesen.

Die Mitarbeiter der University of British Columbia in Kanada haben eine Analyse durchgeführt, mit welchen Werkzeugen die Hybrid Apps umgesetzt wurden. Dazu wurde noch analysiert, für welche Kategorie die Applikation entwickelt wurde. Die Methodik dazu sah folgendermaßen aus: Es wurden Datensätzen von 80.000

Einträgen auf der Open Source Plattform GitHub hergenommen. Sie beinhaltet eine Auflistung von Applikationen, die sowohl für Android als auch für iOS verfügbar sind. Diese Datensätze enthalten nur Attribute zur Applikation und nicht den Quellcode. Dann wurde mit ihren Algorithmus die komplette Android Datenbank durchforstet und überprüft, ob die Applikation in den Datensätzen vorkommt oder nicht. Wenn dieser Datensatz nicht vorgekommen ist, dann wurde die Applikation heruntergeladen und mit ihrem Code die Datei mit den Attributs Einträgen untersucht. Dadurch war ihre Trefferquote bei 100% und es konnte eine saubere Analyse gemacht werden. Der iOS Bereich konnte nicht genau analysiert werden, da die Apps nicht öffentlich verfügbar waren. Es wurden 100 zufällige Applikationen aus dem Datensatz herausgenommen und überprüft. Alle 100 hatten dieselben App Icons und dasselbe User Interface. Um die Analyse zu begrenzen, wurden nur die drei größten Marktführer im Bereich der CPT untersucht. In der Tabelle 2 ist das Ergebnis der Auswertung zu sehen.

Tabelle 2: Auswertung der Marktführer von CPT (Ali & Mesbah, 2016)

CPT	***Anzahl der Apps***	***Prozent der Hybrid Apps***
PhoneGap	*10,562*	*68.0 %*
Titanium	*2,881*	*18.5 %*
Adobe Air	*2,069*	*13.5 %*
Total	***15,512***	***100.00%***

Es ist zu erkennen, dass der Marktführer im Bereich der hybriden Applikation das Framework PhoneGap ist. Außerdem wurde ein weiteres Ergebnis herangeführt um den Standpunkt von Hybrid Apps darzustellen. Das Ziel war es, einen Wert zu finden, um die Qualität von Hybrid Applikation in den diversen App Kategorien darzustellen. Die beiden Gruppen für den Vergleich sind die nativen und hybriden Entwicklungstechnologien. Dazu wurde mit dem mathematischen Ansatz Bayesian Schätzer der Wert für die jeweilige Kategorie der App ermittelt. Unter Kategorien ist das Genre zu verstehen, das zu der Applikation zugewiesen ist. Ein prominentes Einsatzgebiet für diese Theorie ist die Filmdatenbank imdB, die ihre Ratings mit diesem Ansatz erstellt. Dieser Score setzt sich aus Anzahl der Bewertung, Rating, durchschnittliche Anzahl der Bewertung im Datensatz und die durchschnittlichen Ratings im Datensatz, zusammen. Die Bewertung wurde auf Basis der Kategorien erstellt, da es nicht möglich ist, genau die gleiche Applikation zu vergleichen. In der Abbildung 11 ist zu sehen, wie die Bewertungen aussehen: der berechnete Wert wird als AUR (Aggregated User−perceived Rating) bezeichnet.

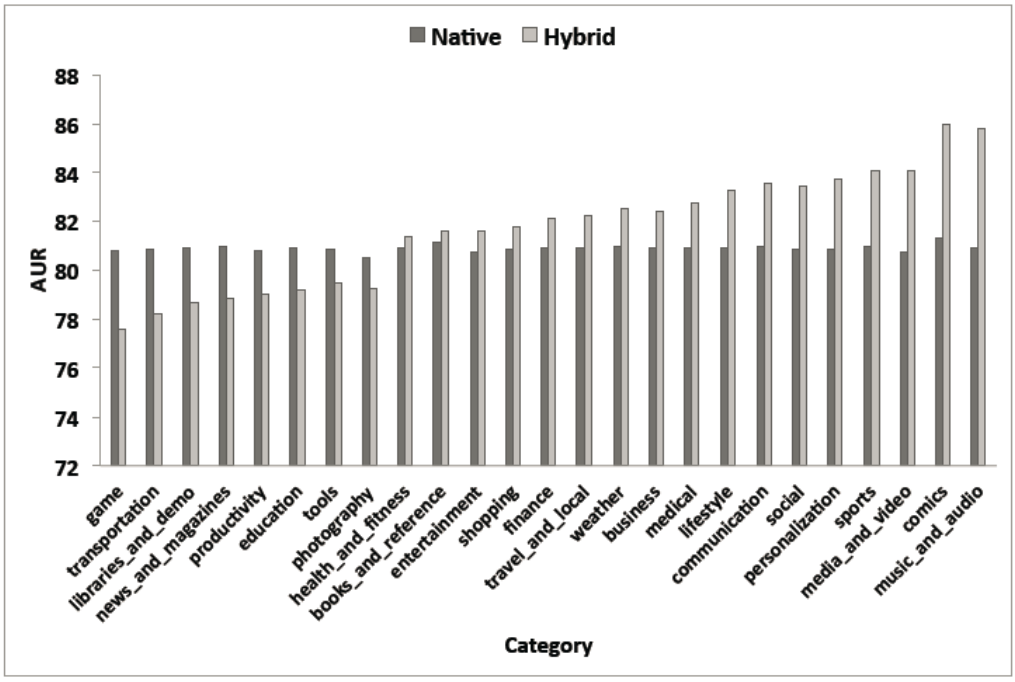


Abbildung 11: Bewertung von Hybrid und Nativen Apps (Ali & Mesbah, 2016)

Außerdem ist zu erkennen, dass die Handhabung mit hybriden Applikationen in vielen Kategorien keine negativen Auswirkungen auf die Bewertung haben. In vielen Kategorien wurden hybride Applikationen sogar besser bewertet als native Apps.

3.2.1 PhoneGap/Cordova

Das Entwicklungstool PhoneGap[12] wurde von der Firma Nitobi entwickelt und im Jahr 2011 von Adobe gekauft. Adobe arbeitet jedoch mit eigenen Entwicklerinnen und Entwicklern weiterhin am Projekt mit und stellt neue Codebeispiele für Benutzerinnen und Benutzer bereit. Für seinen in der Cloud arbeitenden Build-Service PhoneGap Build und die Website für Dokumentation und Beispiele, verwendet das Unternehmen jedoch immer noch den alten Namen. Auch das im Verlauf des Artikels verwendete phonegap-Kommandozeilen-Tool ist nur ein Aufsatz für das eigentliche cordova-CLI von Apache. PhoneGap ist also eine Art Distribution, vergleichbar mit den verschiedenen Linux-Versionen auf dem Markt (Ross, 2013).

PhoneGap besteht aus zwei wesentlichen Bauelementen. Das erste Element ist die JavaScript Bibliothek Cordova, diese stellt die Schnittstelle von HTML, CSS

[12] https://phonegap.com/

und JavaScript zu den Komponenten des Smartphones her. Der zweite Baustein ist die native Engine, sie ermöglicht den Zugriff auf die Sensoren des jeweiligen Betriebssystems. Durch diese nativen Engins könne Apps für die Betriebssysteme Apple iOS, Google Android, LG webOS, Microsoft Windows Phone, Blackberry und Tizen (Adobe, 2017b) erstellt werden. In einer qualitativen Evaluierung wurde eine Testapplikation für Native und Hybrid erstellt. Die evaluierten Einheiten wurden verglichen: Startzeit der Applikation, RAM Verbrauch der Applikation und Dateigröße der App. In allen drei Bereichen schnitt die PhoneGap Applikation besser als die native App ab (Ahti, Hyrynsalmi, & Nevalainen, 2016).

3.2.2 Adobe Air

Das nächste Framework ist auch ein Produkt von Adobe und heißt Adobe Air (Adobe, 2017a). Die Entwicklerzielgruppe ist etwas anders als bei den bereits vorgestellten Frameworks. Hier ist die Zielgruppe 2D und 3D Spiele. Es wurde ein Framework entwickelt, mit dem es möglich ist, plattformübergreifend Spiele zu entwickeln. Dank vollständig GPU-beschleunigten Renderings werden OpenGL- und DirectX-Grafiken in den Hybrid Applikationen perfekt dargestellt. Die Abbildung 12 ist aus einer Studie von der University of British Columbia. Hier wird noch einmal bestätigt, dass der Hauptgebrauch von diesem Framework im Game Bereich liegt.

Das diese Technologien in die Kategorie der hybriden Entwicklungsmethoden eingeordnet wird, liegt an der Architektur des Frameworks. Deswegen wird auch vom Hersteller das Framework dem Bereich der hybriden Technologien zugeordnet.

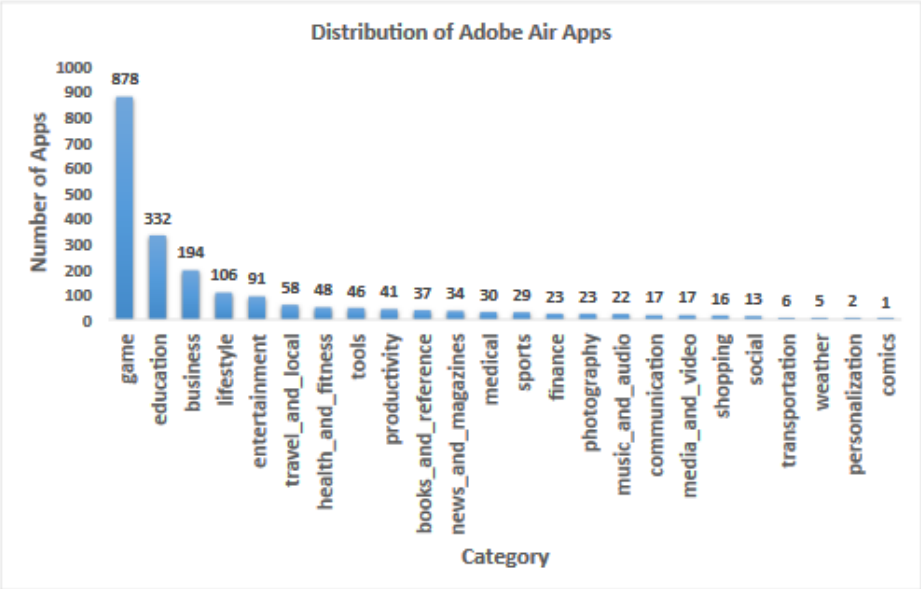


Abbildung 12: Distributionskategorien von Adobe Air Applikationen (Ali & Mesbah, 2016)

3.2.3 React Native

Eine andere Alternative ist das JavaScript Framework **React Native** von Facebook. Das ist keine webbasierte Applikation die einem nativen Container eingebettet wird(Facebook, 2017). Das Framework rendert eine native Applikation mit nativen Komponenten raus. Damit die Rendering Engine ordnungsgemäß läuft, werden die Toolkits wie AndroidSDK oder Apples XCode benötigt. Es wird eine leistungsstarke mobile App, die nicht von einer mit Swift / Objective-C oder Java erstellten App zu unterscheiden ist, erstellt (Milliman, 2017). Auch die großen Flaggschiffe der App Branche haben mitgezogen. Die prominentesten Apps sind Facebook, Instagram[13] und Airbnb. Die User Interface Komponenten müssen für die jeweilige Plattform im Code deklariert werden, aber theoretisch kann die Codefunktionalität gleichbleiben.

3.2.4 Appcelerator Titanium

Appcelerator Titanium ist der zweite große Marktführer im Bereich der Hybrid Applikationen (Appcelerator, 2012). Der Aufbau der Module sieht etwas anders aus als wie bei PhoneGap. Titanium ist eine anwendungsorientierte API für native, mobile und plattformübergreifend Entwicklung. Dabei gibt es aus dem Container eine breite Palette von nativen Features und Funktionen. Das betrifft alle Komponenten der Benutzeroberfläche, als auch Schnittstellen zum System. Die

[13] https://www.instagram.com/

User Interface Elemente und Handhabung sind nativ. Die Rendering Engine rendert die Elemente wie bei React Native in native Komponenten. Das Framework hat eine eigene Entwicklungsumgebung mit der die Software Development Kits (SDK) von Apple und Android verbunden werden können. Zusätzlich gibt es eine Schnittstelle mit der die Benutzerinnen und Benutzer ihre API Schnittstelle konfigurieren können.

3.2.5 Funktionen und Vergleich der Frameworks

Die Funktionspalette von hybriden Applikationen unterscheidet sich nicht eklatant von der nativen Funktionspalette. Durch den nativen Container können auf die Funktionen wie Kamera, Sensoren, Benachrichtigungen und weiteren Funktionen zugegriffen werden. Die Offline Funktionalität wird nicht von jedem hybriden Framework in der Funktionspalette wiedergegeben. Jeder der drei Marktführer im Bereich der hybriden Entwicklungstechnologien fokussieren sich auf unterschiedliche Bereiche.

PhoneGap legt den Fokus auf webtaugliche Applikationen, die in nativen Container verpackt werden. Der große Vorteil ist die Entwicklung von plattformübergreifenden Applikationen. Es werden nicht so viele Ressourcen wie bei anderen Technologien benötigt. Nachteil ist, dass die Probleme von webtauglichen Applikationen mitgenommen werden. Eine Problemstelle ist die Performance, da PhoneGap Apps die Batterieleistung sehr beanspruchen. PhoneGap Applikationen haben auf jeder Plattform dasselbe Layout, das heißt es könnte mit der Usability und Handhabung Probleme geben, weil die Userin und der User mit dem nicht standardisierten User Interface nicht vertraut ist.

Titanium und React Native tauchen mehr in die nativen Bereiche ein und rendern die UI Komponenten nativ. Dadurch hat Titanium und React Native das sogenannte Look & Feel von nativen Applikationen. Auch die Ressourcenaufteilung ist besser und effizienter verteilt, als bei PhoneGap Applikationen, da sich die Architekturen unterscheiden. Ein großer Vorteil ist, dass die Applikationen auch für Offline Aktivitäten programmiert werden können.

Adobe Air ist kein direkter Konkurrent von den anderen Frameworks, da die Spezialisierung nur auf dem Game Bereich liegt. Die Architektur verspricht schnelles und performantes Rendering von 2D und 3D Grafiken.

3.3 Native Entwicklungstechnologien

Die dritte Technologie der großen drei Big Player in den mobilen Entwicklungsoptionen ist die native Entwicklung. Unter der nativen Entwicklung ist

zu verstehen, dass eine Anwendung speziell für eine Plattform entwickelt wird. Das heißt, dieselbe Applikation läuft nicht in identischer Form auf einer anderen Plattform. Darunter ist zu verstehen, dass keine spezielle für iOS entwickelte Applikation auf Android läuft. Die größten drei Betriebssysteme sind iOS von Apple, Android von Google, Windows Phone von Microsoft. Die am meisten eingesetzten Betriebssysteme (Stand Oktober 2017) in Österreich sind Android (64,1 %), iOS (34,6 %) und Windows Phone (1,2 %)(Statista, 2017b). Wie anhand der Statistik zu erkennen, ist Googles Android der Marktführer in Österreich.

Ein Hauptgrund für die Lauffähigkeit auf nur einer speziellen Plattform von nativen Applikationen ist die Programmiersprache. Die unterschiedlichen Plattformen verwenden alle unterschiedliche Programmiersprachen, Android wird in Java programmiert, iOS in Swift und Windows Phone in C# (Apple, 2017b; Google, 2017a; Microsoft, 2017).

Im Themenschwerpunkt der nativen Entwicklungstechnologien werden die Eigenschaften einer nativen Applikation untersucht. Weiters werden die zwei Marktführer iOS und Android näher betrachtet und ihre Interface Richtlinien untersucht und in einem Beispiel verglichen.

3.3.1 Eigenschaften und Funktionalität einer nativen Applikation

Der Zugriff beziehungsweise die Installation läuft über die jeweilige Vertriebsplattform der Betriebssystemhersteller wie Android Play Store oder Apple App Store. Die Benutzerin oder der Benutzer kann ohne großen Aufwand die Applikation im Store suchen und installieren.

Der größte Vorteil einer nativen Applikation ist der Zugriff auf die komplette Funktionspalette. Durch den Code können Kamera, GPS, Bluetooth und andere Sensoren angesteuert werden. So wird für eine bessere Funktionalität oder Nutzererlebnis gesorgt. Push-Benachrichtigungen, Kalender-Zugriff und Touch Gesten können ohne Probleme integriert werden. Die klare und saubere Handhabung in der Applikation von gewohnten Gesten und Funktionen lässt die Userinnen und User sicherer und effizienter mit der App umgehen. Außerdem kann die App auch Offlinemodus laufen, d.h. es wird kein Internetzugang benötigt (McCade, 2017).

Ein negativer Punkt ist: um den kompletten Smartphone Markt abzudecken, müssen mehr Ressourcen verwendet werden als bei anderen Technologien. Diese könnte sich dann zu wirtschaftlichem Nachteil auswirken, wenn der Support für eine Applikation auf mehreren Plattformen organisiert werden muss. Ein weiterer negativer Punkt ist, dass die Applikation für mehrere Geräte entwickelt werden muss, da zum Beispiel bei Apples iPhone die Displaygrößen und Hardwareanforderungen bei vielen Modellen sehr unterschiedlich sind.

Die folgende Eigenschaft ist schwer zu kategorisieren - es ist sowohl Vorteil als auch Nachteil. Wenn eine Applikation in einem Store veröffentlich wird, müssen Richtlinien – oder auch Guidelines genannt - eingehalten werden. Jedes Betriebssystem hat eigene Guidelines, die vorgeben, wie das Styling der Applikation aussehen soll. Der negative Aspekt bei dieser Sache ist, dass wenn der gesamte Smartphone Markt abgedeckt werden soll, müssen mehr Ressourcen in die Entwicklung investiert werden. Es müssen Designs entwickelt und programmiert werden, die mit den Richtlinien der Betriebssystemhersteller kompatibel sind (Meier, 2014).

Ein genauerer Vergleich zwischen den Interface Guidelines von iOS und Android erfolgt in dem Kapitel Interface Guidelines Vergleich iOS und Android.

3.3.2 iOS Betriebssystem

Apples iOS Applikation werden in der Entwicklungsumgebung XCode programmiert. XCode wurde speziell für die Entwicklung von iOS Applikationen designet und entworfen. In der Abbildung 13 wird die Architekturform von iOS dargestellt. Die Schichten bauen aufeinander auf und haben dadurch eine Abhängigkeit.

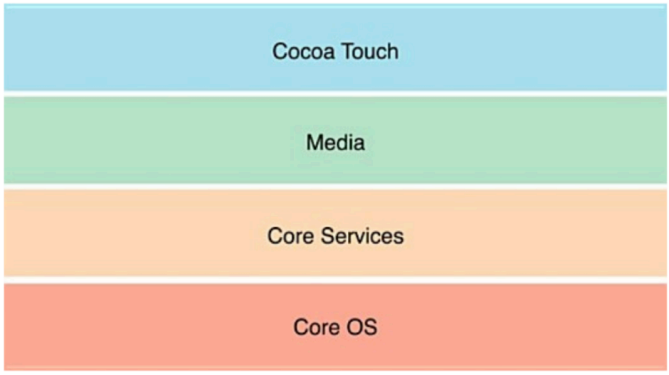


Abbildung 13: iOS Systemarchitektur (Sillmann, 2017)

Core OS behandelt die grundlegenden Funktionalitäten wie Netzwerkkommunikation, Sicherheitsmaßnahmen oder auch andere Hardwarefunktionalitäten. Core Services stellt die Funktionalität zur Verfügung und ermöglicht der Entwicklerin oder den Entwickler den Zugriff zu den Verbindungen von anderen iOS Geräten oder die Speicherverwaltung durch die iCloud. Media ist zuständig für jede Art von multivisuellen Material wie Bilder, Videos, Animation oder weitere Inhalte. Cocoa Touch verbindet alle Schichten und Funktionalitäten, diese Ebene beinhaltet außerdem das UIKit-Framework. Dieses Framework enthält verschiedene Views und Controller für die Entwicklung von iOS Applikationen (Sillmann, 2017).

3.3.3 Android Betriebssystem

Android ist eine Open Source Software die in der Programmiersprache Java entwickelt wird. Die Entwicklung kann in einer Java Umgebung oder in Android Studio stattfinden. Das Android Betriebssystem wird auf einem Linux Kernel aufgebaut, der die Schnittstelle zur Hardware ist. Zusätzlich wurden im System Entwicklungsoptionen verknüpft, die der Programmiererin bzw. dem Programmierer es möglich macht, auf Open Source Bibliotheken wie SQLite, WebKit oder OpenGL zuzugreifen.

Android stellt das Software Development Kit (SDK) den Entwicklerinnen und Entwickler zur Verfügung. Das SDK beinhaltet eine große Funktionspalette. Dadurch ist auf Anwendungsebene sämtliche Funktionalität wie Standortverwaltung, Datenbankzugriff oder andere Hardwarefunktionalitäten gegeben (Meier, 2014).

3.3.4 Interface Guidelines Vergleich iOS und Android

Die beiden Marktführer haben in ihren Dokumentationen Richtlinien festgelegt, damit die Entwicklerinnen und Entwickler eine Grundlage für die Designprinzipien der Applikation haben. In einem Bespiel werden die Unterschiede von iOS und Android aufgezeigt.

Bei den Interface Guidelines von Apples iOS wurden folgende Standards definiert, es soll in der Applikation die Möglichkeit gegeben sein, um zum vorigen Bildschirm oder Screen zu gelangen. Bei Android Smartphones ist oft der universelle Zurück Button gegeben. Bei der Apple Hardware ist das etwas anders, hier muss in der sogenannten Navigationsleiste ein Zurück Button implementiert sein (Apple, 2018).

In Abbildung 14 wird ein Beispiel mit einer sauberen Implementierung eines Zurück Buttons in einer iOS Applikation gezeigt.

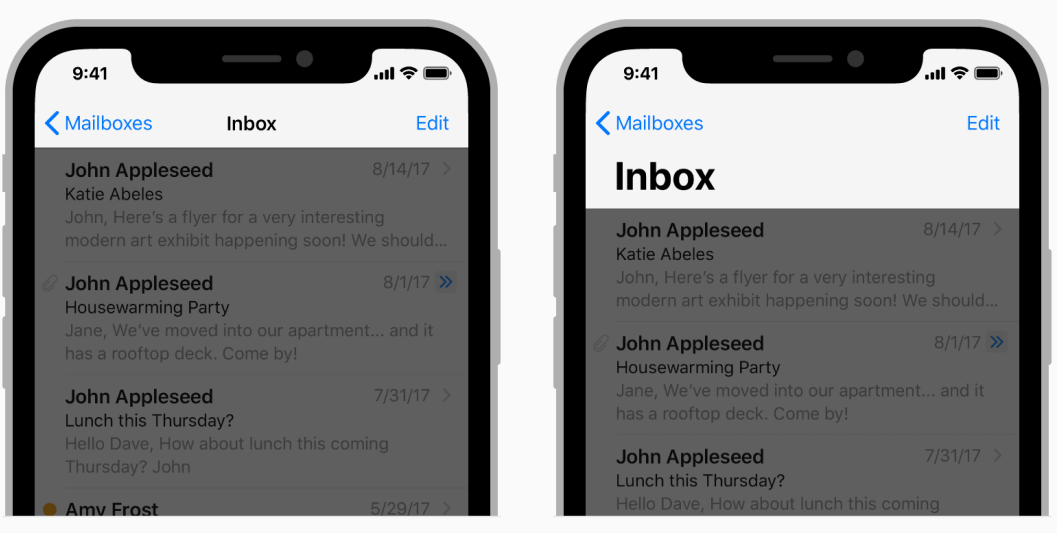


Abbildung 14: iOS Guidelines für die Navigationsleiste (Apple, 2018)

Bei Android Geräten werden auf die Interface Guidelines folgende Richtlinien definiert: Die Option dazu ist, entweder mit dem physischen *Zurück Button* am mobilen Gerät oder auch wie bei iOS in der Navigationsleiste. In der Abbildung 15 werden beide Möglichkeiten definiert, anderes als bei iOS ist es keine Vorrausetzung den *Zurück Button* in der Navigationsleiste zu definieren.

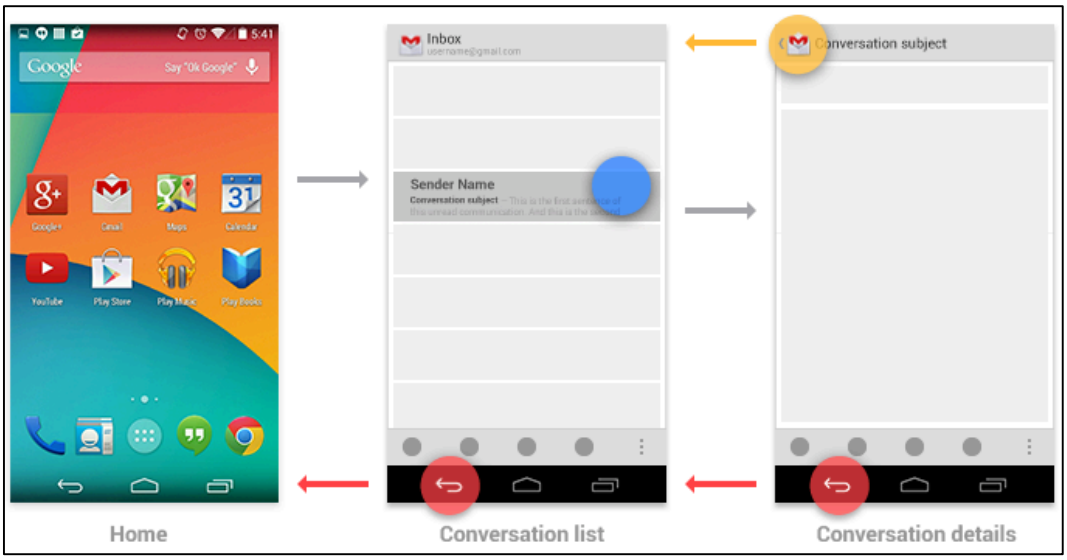


Abbildung 15: Android Guidelines für die Navigationsleiste (Google, 2018)

Der positive Aspekt bei den Guidelines ist, dass durch diese Richtlinien die Benutzerin oder der Benutzer bereits Erfahrungen mit der Interaktion hat. So weiß die Anwenderin oder der Anwender z.B., wenn man auf dem iPhone zurück zum vorigen Screen kommen möchte, muss auf den Button links oben geklickt werden, aber auch mit einer Wischbewegung nach rechts, wäre dies möglich. Wenn diese Richtlinien nicht gegeben wären, könnte die Implementierung dieser Funktionen

anders aussehen und die Userin oder der User würden anders mit den Applikationen interagieren. Dadurch könnten negative Effekte wie Unzufriedenheit entstehen.

Ein weiterer positiver Effekt der Guidelines ist es, wenn die Applikation dann im Store hochgeladen wird, gibt es bei Apple den Review Prozess und bei Android Launch Prozess. Hier wird die Applikation auf Sicherheitslücken und Qualitätsrichtlinien untersucht (Apple, 2017a; Google, 2017b). Android hat eine Checkliste veröffentlicht, die noch kontrolliert werden sollte bevor die App für den Launch Prozess freigegeben wird. Der Review Prozess der Android Applikationen hat eine durchschnittliche Dauer von 48 Stunden, bei Apple dauert es ca. um die 8 Tage bis die Applikation im Store erscheint. Diese Review Prozesse sind von Apple und Android eingerichtet worden, damit die Stores nicht mit Spam oder Viren Apps überlaufen werden.

3.3.5 Fazit

Zusammenfassend ist kann festgestellt werden, dass native Entwicklungsmethoden den performantesten Zugang zu Hardwareschnittstellen haben. Entwicklerinnen und Entwickler sollten beachten, wenn der größte Teil des mobilen Marktes abdeckt werden soll, müssen weitere Parameter bedacht werden. Ein Beispiel dafür ist, dass mehrere Programmiersprachen benötigt werden um iOS und Android Applikationen umzusetzen. Außerdem muss auch bei der Entwicklung des Interface Designs auf die Guidelines Rücksicht genommen werden.

3.4 Vergleich der Entwicklungsmethoden

In diesem Themenbereich wird der Vergleich bzw. die Gegenüberstellung der mobilen Entwicklungsmethoden dargestellt. Der Ausgangspunkt ist die native Funktionsplatte. Jede der einzelnen Plattformen bietet Vor- und Nachteile. In der

Tabelle *3* werden die Funktionalitäten verglichen.

Tabelle 3: Direkter Vergleich der drei Entwicklungsoptionen

Native Apps	***Hybrid Apps***	***Web Apps***
Zugriff auf alle Hardwarefunktionen wie Sensoren	*Mittlerweile decken Cordova und PhoneGap fast komplette Hardwarefunktionalität ab*	*Eingeschränkter Zugriff auf die Low-Level Hardwarefunktionen wie Standort, Push- Berichtigungen oder Kamera*
Benutzbar im Offlinemodus	*Benutzbar im Offlinemodus*	*Durch die Integration von technischen Lösungen wie Service Worker auch im Offlinemodus benutzbar*
Installation durch die Distributionsplattform wie App Store oder Play Store	*Installation durch die Distributionsplattform wie App Store oder Play Store*	*Verwendbarkeit durch den mobilen Browser, Einsatz von Cordova möglich*
Unterschiedliche Programmiersprachen für die jeweilige Plattform.	*Verwendung von Web Standards wie HTML, CSS und als Programmiersprache JavaScript*	*Verwendung von Web Standards wie HTML, CSS und JavaScript*
Verwendbarkeit nur auf jeweiligen Plattform	*Gleiche Codebasis für verschiedene Plattformen.*	*Plattformunabhängig*
Aktualisierungen müssen über die Distributionsplattform installiert werden. Verfügbar erst nach der Durchführung der Updates	*Aktualisierungen müssen über die Distributionsplattform installiert werden. Verfügbar erst nach der Durchführung der Updates*	*Aktualisierungen sind durch erneuten Aufruf der Web Applikation verfügbar*
Top Performance, durch direkten Zugriff auf die Hardwaremodule	*Je nach Wahl des Frameworks gibt es Performance Einschränkungen*	*Performanceverlust aber kaum wahrnehmbar, alternative Lösungen sind Service Worker*
User Interface wird mit nativen Element gebaut	*User Interface wird mit Web Technologien nachgebaut*	*User Interface wird mit Web Technologien nachgebaut*
User Interface wird nach den Style Richtlinien der jeweiligen Plattformen gebaut	*Keine Richtlinien bei der Gestaltung des UI, möglicherweise kann eine Problematik bei dem Review Prozess durch die Vertriebsplattform entstehen*	*Keine Richtlinien bei der Gestaltung des UI*
Ideale Umgebung für die Entwicklung von Spielen	*Eingeschränkte Umgebung und Performanceverluste*	*Eingeschränkte Umgebung und Performanceverluste da das Spiel für den Browser entwickelt werden muss*
Volle Funktionalität bei der Verwendung der Touch Gesten	*Volle Funktionalität bei der Verwendung der Touch Gesten*	*Eingeschränkte Funktionalität bei der Verwendung der Touch Gesten*

Wie zu erkennen ist, gibt es in jeder der mobilen Entwicklungsmethoden Vor- und Nachteile. Die Hauptkriterien für die Auswahl der passenden Methodik sollten das Anwendungsgebiet und das nötige Know-How jeweilige Entwicklungsoption sein. In meisten Fällen gibt es immer einen alternativen Weg wie man eine Funktion abbilden kann, die von der ausgewählten Entwicklungsoption nicht mitgeliefert wird.

Ein Fallbeispiel dazu wäre: die Entwicklerin oder der Entwickler möchten eine iOS Applikation entwickeln und für die Authentifizierung den Fingerprint Sensor verwenden. Das erlernte Wissen ist aber im Bereich der Webtechnologien, deshalb wäre die Wahl eines Hybrid Frameworks mit dem benötigten Cordova Plugin für den Fingerprint Sensor die beste Wahl für die Faktoren Zeit und Wirtschaftlichkeit. In der Studie wurde genau dieser Punkt angeschnitten und es wurden Befragungen durchgeführt, warum Entwicklerinnen und Entwickler mit hybriden Frameworks arbeiten (Mohammadi Kho'i & Jahid, 2016). Hauptsächlich war entscheidend, dass die Eintrittsbarrieren bei hybriden Entwicklungsmethoden geringer, als bei nativen Entwicklungsmethoden sind.

Tabelle 4: Performance Test von nativen und hybriden Applikationen (Corral, Sillitti, & Succi, 2012)

Parameter	***Native App***	***Hybrid App***
Zugriff Beschleunigungssensor	*0.7136*	*32608*
Sound Benachrichtigung	*18.4835*	*26.7481*
Ausführung Vibration	*1.5134*	*3.2222*
Daten empfangen vom GPS Signal	*2.1881*	*809.2352*
Daten empfangen aus dem Netzwerk	*1.1015*	*1.01419*
Schreibzugriffe	*4.7146*	*7.9221*
Lesezugriffe	*13.3036*	*255.7381*
Daten empfangen aus der Kontaktliste	*95.8686*	*1841.4689*

Ein markanter Punkt im Vergleich zwischen Nativen Apps und Hybrid Apps ist die Performance. Mit diesem Thema befasste sich der Artikel „Mobile Multiplatform Development: An Experiment for Performance Analysis" beschäftigt (Corral u. a.,

2012). In Tabelle 4 gibt es verschiedene Anwendungsgebiete in der die Performance Tests durchgeführt wurden. Die dargestellten Werte werden in Millisekunden angegeben.

Wie zu erkennen ist, liegt die Performance der nativen Methodik in allen Bereichen, außer dem Netzwerkzugriff, vor der hybriden Methodik. Diese Performancetests wurden in dem Jahr 2012 durchgeführt. Heutzutage haben die mobilen Endgeräte bessere Hardware und würden noch schnellere Zugriffe liefern.

4 Problemanalyse

Um den aktuellen Prozess und das Anforderungsprofil für eine Applikation für Brückeninspektionen zu definieren, wurde ein Interview mit einer Expertin durchgeführt und bestehende Dokumente im Bereich der Brückeninspektion analysiert. In diesem Kapitel wurden die Methodik, Durchführung, Ansätze und Resultate bewertet und diskutiert.

4.1 Expertinnen Interview

Es wurde ein Interview mit einer Expertin durchgeführt. Sie ist Researcherin am Department Bahntechnologie und Mobilität an der Fachhochschule in St. Pölten. Sie hat zahlreiche Universitätsabschlüsse in den Bereichen Landmanagement, Infrastruktur und Bautechnik, Kulturtechnik und Wasserwirtschaft, Wasserwirtschaft und - Umwelt (FH St.Pölten, 2017). Außerdem hat die Expertin Einblicke in den Bereichen Abläufe, Instrumente und Durchführung der Brückeninspektion.

Diese Form von Interview wurde gewählt, um die Abläufe einer Brückeninspektion kennenzulernen und die Systematik dahinter zu erfassen.

4.1.1 Leitfaden

Der Leitfaden für das Interview wurde in drei Teilen aufgeteilt: der erste Teil beschäftigte sich mit der Brückeninspektion. Es wurden allgemeine Fragen über das Thema Brückeninspektion gestellt. Der zweite Teil beschäftige sich intensiver mit der Materie Brückeninspektion und der Fokus des Interviews lag mehr auf Detailfragen über diverse Abläufe. Zusätzlich wurden Fachbegriffe und Dokumente von der Interviewpartnerin erläutert. Der dritte Teil behandelte die bekannten Problematiken und möglichen Spezifikationen für Prozessoptimierungen.

4.1.2 Durchführung

Das Interview wurde am 09. September 2017 in Büroräumen des BIZ St.Pölten durchgeführt. Die Aufzeichnung des Interviews erfolgte in Form von handschriftlichen Notizen. Das Interview dauerte ca. 40 Minuten.

4.1.3 Anforderungserhebung

Anlässlich des Interviews mit der Expertin im Bereich Bahntechnologie und Mobilität wurde ein Anforderungsprofil erstellt. Die Anforderungserhebung ist der Grundbaustein für die Entwicklung des späteren Anforderungsprofils, das nach der Auswertung des Interviews erstellt wurde. Die Gliederung des Anforderungsprofils wird nach der Anordnung der Prioritäten der Auswertung erfolgen.

4.1.3.1 Inspektionsprozess

Die Interviewpartnerin wurde befragt, wie der derzeitige Inspektionsprozess abläuft. Wichtigster Kernpunkt bei dieser Fragestellung war, wie der genaue Ablauf bei einer Brückeninspektion aussieht. Ziel der Auswertung war es, eine Übersicht über den Inspektionsprozess zu bekommen. Die genauen Prozessbeschreibungen wurden durch Detailfragen zu späteren Punkten erörtert.

1. Das ausgewählte Fachpersonal wird für eine Brückeninspektion oder für ein Bauobjekt zugewiesen. Das Fachpersonal leitet, untersucht und beurteilt die Brücken bzw. das Bauobjekt.
2. Um die Inspektion korrekt durchzuführen, muss eine ordnungsgemäße Vorbereitung stattfinden. Dazu ist die Durchsicht von vorherigen Inspektionsprotokollen notwendig.
3. Der Inspektionstermin wird festgeltet und zusätzlich wird Rücksicht auf die Wettersituation genommen. Wichtige Voraussetzung bei der Inspektion sind gute Lichtverhältnisse für Besichtigungen und Fotodokumentation.
4. Vorbereitung von Hilfsmitteln und Werkzeugen, die für die Inspektion gebraucht und verwendet werden.
5. Untersuchen der Brücke bzw. des Bauobjektes - Augenmerk wird auf Risse, Abbröckelungen, Korrosion und weitere Mängel gelegt.
6. Durchführung diverser Prüfungsmethoden und Standards wie zum Beispiel: die Richtlinien der österreichischen Bundesbahnen für Brückeninspektion.
7. Dokumentation von Schäden und Mängeln, zusätzlich erfolgt die Dokumentation mit Fotomaterial. Die Schadensdokumentation muss gemäß eines Protokolls abgearbeitet werden.
8. Falls nötige Maßnahmen getroffen werden müssen, weil Schäden irreparable sind, werden Angaben dazu gemacht.
9. Vor dem Abschluss der Inspektion wird noch eine Frist für die nächste Brückeninspektion bzw. -kontrolle angesetzt.
10. Die Inspektionsprotokolle und Unterlagen werden dann von der jeweiligen Behörde archiviert.

4.1.3.2 Auswertung

Durch das Expertinnen Interview wurde der Ablauf und Prozess einer Brückeninspektion aus dem Blickwinkel einer Expertin definiert. Diese Kernpunkte und Details sind Richtlinien für das Anforderungsprofil der Applikation. Zusätzlich wurden einige Problemstellungen beim Interview erläutert. Hier wird jetzt genauer auf die folgenden Problemstellen eingegangen.

4.1.3.2.1 Dokumentation und Zugriff

Die Dokumentation einer Brückeninspektion findet handschriftlich auf dem sogenannten Revisionsblatt statt. Auf dieses Formblatt wird später in der Anforderungserhebung der Dokumente eingegangen. Am Revisionsblatt werden die Schäden und Mängel dokumentiert, zusätzlich wird Bildmaterial aufgenommen. Die Dokumentation und das Bildmaterial werden bei der jeweiligen zuständigen Behörde archiviert. Durch diese Methode könnten einige Probleme entstehen, wie z.B. die mögliche Fehlerquote bei der handschriftlichen Dokumentation. Die durchgeführte Dokumentation am Revisionsblatt erstreckt sich meist über einen jahrelangen Zeitraum. Jeder der Inspektorinnen und Inspektoren hat eine andere Handschrift und einen anderen Stil, z.B. durch Unstimmigkeiten in Form von Lesbarkeit und Ausdrucksweise. Dies könnte für die nachfolgenden Brückeninspektorinnen und -inspektoren Probleme bereiten und wichtige Textstellen nicht genau lesbar sein, oder auch nicht verstanden werden. Notwendige Information könnten verloren gehen. Durch diese Problematik könnten wirtschaftliche Folgen wie Zusatzinspektionen oder nicht benötigte Sanierungsarbeiten entstehen. Eine weitere Problematik ist die Dokumentenverwaltung, denn es gibt kein zentrales Datenverzeichnis oder eine Datenbank, die über Internet abrufbar ist. Das bedeutet eine sehr geringe Flexibilität und weitere nicht genutzte Ressourcen in Form von Wartezeiten. Für einen Zugriff auf benötigte Dokumente muss bei den zuständigen Behörden angefragt werden und Inspektionsmaterial aus den Archiven beschafft werden.

4.1.3.2.2 Anwendungsfälle

Es gibt auch spezielle Anwendungsfälle, die man kaum beeinflussen kann. Wenn Inspektionen bei nicht optimalen Wetterbedingungen durchgeführt werden, müssen eventuell die Revisionsblätter bei Schneefall oder Regen ausgefüllt werden. Papier ist nicht für alle Witterungsverhältnisse ideal und die Dokumentation wird dann nicht ordnungsgemäß geschützt.

4.1.3.2.3 Prozessoptimierung

Im Expertinnen Interview wurde auch das Thema Anmerkungen angeschnitten: die Expertin erklärte ihre Sicht und auch mögliche Prozessverbesserungen. Ein Thema bzw. der Ausgangspunkt war, wie man den Prozess der Brückeninspektion

mit mobilen Geräten verbessern könnte. Die Expertin hat interessante Ansätze, Ideen und dazu mögliche Unzulänglichkeiten genannt.

4.1.3.2.4 Fotodokumentation

Behörden und Unternehmen, die Brückeninspektionen durchführen, haben bei den Inspektionen bereits mobile Geräte wie Smartphones oder Tablets dabei. Die Fotodokumentation findet dann mit dem mobilen Gerät oder mit einer Kompaktkamera bzw. Spiegelreflexkamera statt. Die Thematik um die Fotoqualität von mobilen Geräten sollte kein Problempunkt werden, denn viele Smartphones haben bereits sehr hochwertige Kameramodule in den Geräten eingebaut. Aber der Tablet Markt hinkt in diesem Segment etwas hinterher. Wie bereits in der Auswertung erwähnt, werden auch Dokumentationen bei nicht idealen Wetterverhältnissen oder Lichtverhältnissen durchgeführt. Dadurch könnten bei schlechten Lichtverhältnissen mit mobilen Geräten Qualitätsmängel entstehen. Ein positiver Aspekt ist, dass die meisten mobilen Geräte über eine Panoramafunktion verfügen und daher die Abdeckung eines größeren Betrachtungswinkels ermöglichen. Durch diese Funktion können die Anwenderin und der Anwender die Bauwerke bzw. Brücken komplett abbilden.

Ein zusätzlicher positiver Aspekt ist es, dass durch eine mögliche Funktionalität, die Markierung von Schäden am Fotomaterial angezeigt werden kann. Der derzeitige Prozess dazu sieht folgendermaßen aus:

1. Inspektorin oder Inspektor erstellt ein Foto vom Bauobjekt bzw. Brücke.
2. Die Fotos werden dann im Büro der Behörde oder der Firma, die Inspektion leitet, ausgedruckt.
3. Dann erfolgen die Schadensmarkierungen am Fotomaterial und die Beschriftung handschriftlich auf einem zusätzlichen Dokument.

Mobile Technologien besitzen bereits die Möglichkeit, diese Funktionalität technisch abzubilden. Ein zusätzlicher Ansatz wäre es, dass auch Schäden, wie Risse oder andere Mängel gleich markiert werden könnten. Außerdem kann auch die nötige Beschreibung des Schadens bzw. Notizen gleich dokumentiert werden.

4.1.3.2.5 Planungsmaterial

Durch den Einsatz eines mobilen Gerätes könnten die Übersichtspläne der Inspektionsobjekte gleich mitgeführt werden. Ein weiterer Ansatz wäre das Fotomaterial gleich mit dem Übersichtsplan zu kombinieren. Die Expertin hat die Idee dahinter so definiert:

1. Am Übersichtsplan kann ein Schaden gleich markiert werden. Zusätzlich kann dazu eine Abbildung des Schadens hochgeladen und mit der markierten Stelle verknüpft werden.

2. Eine weitere Möglichkeit wäre, den Schaden gleich mit zusätzlichen Parametern anzupassen. Beispiele dafür wären Klassifizierung oder die Beschreibung der Abbildung.

Ein negativer Aspekt hinter diesem Ansatz ist die Verwaltung von vorhandenem Material, denn in vielen Gemeinden gibt es keine Detailpläne zu den Brücken und Bauobjekten.

4.1.3.2.6 Weitere Dokumentationsmöglichkeiten

Falls zu Bauobjekten Pläne vorhanden sind, werden derzeit die mobilen Geräte bei Inspektionen eingesetzt. Weitere Ansätze und Konzepte für die Dokumentation haben sich in den letzten Jahren entwickelt, denn je nach Brücke gibt es verschiedene Standards und Normen um Schäden zu klassifizieren. Ein interessanter Konzeptansatz wäre es, wenn die möglichen Schäden gleich beschrieben und die Klassifizierungskennzeichnung der verwendeten Norm eingearbeitet wird. Die Inspektionsberichte werden bei der jeweiligen Behörde gleich archiviert. Eine weitere brauchbare Nuance wäre es, gleich den Inspektionsbericht mit technischen Möglichkeiten als PDF Datei zu generieren. PDF (Portable Document Format) ist ein Dateiformat, das die Präsentation und den Austausch von Dokumenten unabhängig von Original-Software, Hardware oder Betriebssystem ermöglicht (Adobe, 2017c)

Die Berichte sollen auf zwei verschiedene Arten archiviert werden: auf der einen Seite in Papierform bei der jeweiligen zuständigen Inspektionsbehörde. Der zweite Weg soll in digitaler Form stattfinden, die Inspektionsdaten sollen auf einem Server abgelegt werden. Ein wichtiger Punkt der beachtet werden muss, sind die Erfüllung der nötigen Schritte und Richtlinien sowie die Gewährleistung der höchsten Datensicherheit.

4.2 Anforderungserhebung Dokumentenanalyse

Die Anforderungsanalyse besteht aus zwei Haupteilen: die Anforderungserhebung aus dem Expertinnen Interview und die Anforderungserhebung aus der Dokumentenanalyse. Bei der Dokumentenanalyse werden Dokumente von den Österreichischen Bundesbahnen und das Dokument Bauwerksprüfung nach DIN 1076 aus deutschen Bundesministerium für Verkehr, Bau und Stadtentwicklung herangezogen. Um einen Vergleich zwischen den Standards für Brückeninspektionen verschiedener Länder darzustellen, wurde das deutsche Dokument in die Analyse aufgenommen.

Außerdem wird das ÖBB Revisionsblatt, das bei Brückeninspektionen eingesetzt wird, analysiert und näher auf die einzelnen Eigenschaften eingegangen.

4.2.1 Ansatz der Analyse

Um die verschiedenen Dokumente mit den gleichen Anforderungen zu analysieren, müssen verschiedene Kriterien definiert sein. Durch die Definition der einzelnen Kriterien können die Dokumente nach der gleichen Struktur analysiert werden. Die Auflistung der Kriterien sieht folgendermaßen aus:

1. **Gliederung und Aufteilung des Prozesses bzw. der Teilprozesse:** Wie wird der Brückeninspektionsprozess in den einzelnen Dokumenten definiert und was sind seine Eigenschaften? Wie sieht die Gliederung aus und wie wurden die Teilprozesse definiert?
2. **Klassifizierung von Schadenskategorien:** Wenn es Schadensklassifizierungen gibt, wie werden diese definiert und abgebildet?
3. **Zusammenfassung:** Welche der Eigenschaften sind für die Anforderungsanalyse relevant?

Die Analysekriterien werden nicht für das Revisionsblatt angewendet, da dieses Dokument für diese Kriterien keine Relevanz besitzt. Das Revisionsblatt ist ein Teildokument, das für die Verwaltung von Inspektionsdaten der Österreichischen Bundesbahnen dient. Das Dokument wird auf seine Eigenschaften und Charakteristiken geprüft und analysiert.

4.2.2 Dokument ÖBB: Richtlinie für die Überwachung und Inspektion von Eisenbahnbrücken in Spannbetonweise

Dieses Dokument wurde in Zusammenarbeit von den ÖBB und dem Unternehmen Zivilingenieure für Bauwesen Kirsch - Muchitsch und Partner erarbeitet und erstellt (Österreichische Bundesbahnen, 1996).

Das Dokument wurde 1996 erstellt und ist noch immer Standard bei Brückeninspektionen für Spannbetonweise bei den ÖBB.

4.2.2.1 Gliederung und Aufteilung der Teilprozesse

Wie der Ablauf einer Inspektion aussieht wurde ziemlich genau und klar in ÖBB-Richtlinien formuliert. Die Aufteilung der Inspektionsprozesse wurde in zwei Teile gegliedert, die folgendermaßen lauten: **Spannbetonbrückenüberwachung Allgemeiner Teil** und **Spannbetonbrückenüberwachung Inspektionshilfen**. Die beinhalten die einzelnen Subprozesse, die Analyse der Gliederung beginnt mit dem allgemeinen Teil der Gliederung.

4.2.2.1.1 Allgemein – Einführung

Der Einführungsprozess deckt Vorbereitungsmaßnahmen, Richtlinien und weitere Anforderungen ab.

Jede Brückeninspektion wird nach einer gewissen Norm durchgeführt. Deswegen ist es für die Inspektorinnen und Inspektoren wichtig, welche Norm und Regelwerk für die Inspektion relevant ist. In diesen ÖBB-Richtlinien sind folgende Normen angegeben:

- ÖNORM B 4253: Norm für Spannbetonbrücken-Eisenbahnbrücken, Berechnung und Ausführung
- ÖNORM B 4703: Norm für Berechnung und Ausführung der Tragwerke, Massivbau, Eisenbahnbrücken

Außerdem ist definiert, von welchem Personal Brückeninspektionen durchgeführt werden dürfen. Folgendes Personal darf Brückeninspektionen durchführen:

- Fachleute, die in der Brückenerhaltung tätig sind und demnach auch die Überwachung und Inspektion von Spannbetontragwerken anordnen, leiten und schlussendlich beurteilen, müssen im Spannbetonbrückenbau sowohl theoretische (statische und konstruktive), wie auch baupraktische Erfahrungen haben und die vorweg genannten Regelwerke kennen (Österreichische Bundesbahnen, 1996).
- Fachpersonal, das die Inspektion vor Ort durchführt wie PolierInnen, BauleiterInnen, KonstrukteurInnen, SpannmeisterInnen und weiteres Fachpersonal.

Eine weitere Anforderung im Einführungsprozess ist die Vorbereitungsphase. Damit eine Inspektion stattfinden kann, müssen diverse Vorbereitungen getroffen werden. Dazu zählen folgende Aufgabenbereiche:

- Organisation von Unterlagen wie Protokolle, Berichte, Ergebnisse, sämtliches Planungsmaterial und andere diverser Unterlagen.
- Kritische Durchsicht dieser gesammelten Unterlagen, es soll eine Übersicht von möglichen Problemstellen geschaffen werden. Dieser Bereich hat eine Besonderheit, weil durch diese Problemstellen der Schwerpunkt der Inspektion gesetzt wird.
- Außerdem muss das Inspektionsteam besetzt und das Aufgabengebiet eingeteilt werden.
- Wenn das Objekt nicht genau bekannt ist, empfiehlt sich eine kurze Vorbesichtigung zur Erkundung der Zufahrt, der Gesamtsituation, der Zugänglichkeit und zur Festlegung von notwendigen Vorarbeiten (Österreichische Bundesbahnen, 1996).
- Der Inspektionstermin muss vorher angesetzt werden und sollte bei optimalen Wetterbedingungen durchgeführt werden. Die Hauptgründe dafür sind, dass die Gefahren bei der Inspektion für das Team minimiert und die Qualität der Inspektion gesteigert wird. Damit das Fotomaterial auf

einem hohen Level gehalten wird, sollten guten Lichtverhältnisse herrschen.

Wenn diese Anforderungen erfüllt sind, müssen bei manchen Szenarien noch Vorbereitungen am Bauobjekt vorgenommen werden. Folgende Maßnahmen müssen getroffen werden, bevor die Inspektion beginnt:

- Inspektionsumgebung sollte so präpariert werden, damit eine ordnungsgemäße Inspektion durchgeführt werden kann. Zufahrten sollten gut erreichbar sein und die Bauteile frei von Gestrüpp und hohem Gras.
- Um das Inspektionsteam in keine Gefahrensituationen zu bringen, sollte der Inspektionsbereich durch Gefährdungen wie Zugsverkehr und Oberleitungen abgesichert werden.
- Reinigung von verschmutzten Bauteile mit Druckwasser, Druckluft, Besen und Drahtbürste. Gelockerte und abgesprengte Betonteile entfernen und allenfalls aufbewahren(Österreichische Bundesbahnen, 1996).

Bei einer Brückeninspektion zählen auch zur Vorbereitung die ordnungsgemäße Verwaltung des Werkzeuges und der Geräte, die für die Inspektion benötigt werden. Werkzeuge wie Geologenhammer, Winkelmesser, Betonprüfhammer und diverse andere Werkzeuge werden für die Inspektion benötigt. Außerdem muss auch eine geeignete Kamera mit Blitzlicht in das Equipment für die Inspektion aufgenommen werden. In der heutigen Zeit werden hochauflösende Kameras bei Inspektionen verwendet. Darunter fallen Geräte wie Smartphones, Tablets oder Spiegelreflexkameras.

Die Vorbereitungsmaßnahmen decken einen großen Themenbereich ab und sind sorgfältig zu erledigen, damit später eine ordnungsgemäße Brückeninspektion durchgeführt werden kann. Zusätzlich haben diese Maßnahmen eine sehr hohe Priorität für die Sicherheit des Inspektionsteams.

4.2.2.1.2 Allgemein – Inspektion vor Ort

Nach der Vorbereitungsphase kommt die Inspektionsphase: der nächste Teilprozess ist die Inspektion vor Ort. In diesem Teilprozess wird erläutert, welche Aktionen bei einer Inspektion durchgeführt werden müssen.

Wenn alle Vorbereitungen organisiert wurden, sieht eine Brückeninspektion folgendermaßen aus:

- Ziel ist, alle Tragwerke und Lagerzonen an allen Stellen so nah wie möglich zu erreichen. Es müssen die Betonstellen mit dem Geologenhammer abgeklopft werden. Zusätzlich muss die Inspektorin oder der Inspektor auch andere Problemstellen - wie feine Risse - im Auge

behalten. Wenn es nicht möglich ist, nah genug an die Problemstellen zu gelangen, müssen die Zonen mit einem Fernglas kontrolliert werden.

- Wenn Fehler oder Mängel bei der Brückeninspektion auftreten, müssen diese dann nach dem Standard der Spannbetonweise für Eisenbahnbrücken dokumentiert werden.

Welche Richtlinien und Vorrausetzungen die Dokumentation hat, wird im nachfolgenden Punkt ausführlich erklärt.

4.2.2.1.3 Allgemein – Dokumentation und Bewertung

Der nächste Prozess ist der Dokumentationsprozess: hier werden alle Schäden und Mängel, die bei der Inspektion aufgenommen werden, dokumentiert. Wichtige Elemente der Dokumentation sind die Kurzbeschreibung und das Symbol für die dazugehörige Schadenskategorie. In den Richtlinien der ÖBB wird nachfolgendem Schema vorgegangen:

- Welcher Inspektor oder Inspektorin hat inspiziert? Wann wurde die Inspektion durchgeführt und unter welchen Bedingungen?
- Kurzbeschreibung des Bauwerkes (gleichbleibend von letzter Hauptprüfung oder zwischenzeitliche Umbaumaßnahmen einfließen lassen) (Österreichische Bundesbahnen, 1996).
- Dokumentieren in diversen Listen und Plänen wie Schadensliste und Übersichtplan.
- Zusätzlich zur Schadensdokumentation erfolgt die Dokumentation durch Fotomaterial.

In den Schadenslisten werden die Schäden und Mängel in folgende Elemente gegliedert:

- Kurzbeschreibung, hier wird im Kurztext der Schaden oder Mangel angemerkt.
- Die Schadenskategorie mit dem dazugehörigen Symbol und Anhang der Fotodokumentation.
- Wie die Schadenskategorie klassifiziert und zugeordnet wird, bestimmt die Inspektionsleiterin oder der Inspektionsleiter bei einer Gutachtervisite.
- Die Skizzierung ist eine zusätzliche Methodik um die Mängel besser in der Dokumentation beschreiben zu können

Wie die Norm, die Schadensymbole der einzelnen Kategorien handhabt und wie die Definition von Schadensklassen aussieht, wird in dem Kapitel Klassifizierung von Schadenskategorien erklärt und beschrieben.

In der Prozessgliederung gibt es auch den Teilprozess der Sonderprüfungen: diese Sonderprüfungen werden bei speziellen Fehlermöglichkeiten eingesetzt.

Grundsätzlich unterscheidet sich eine Spannbetonbrücke nicht in vielen Bereichen von einer *schlaff bewehrten Brücke*. Es gibt jedoch in Spannbetontragwerken Bereiche, die eine besonders aufmerksame Durchsicht verlangen und es müssen grundsätzlich alle Mängel auf ihren möglichen Zusammenhang mit der Spannbetonbauweise und den Spanngliedern hin überlegt werden (Österreichische Bundesbahnen, 1996).

Die Expertin oder der Experte erkennt bei der Durchsicht der Bestandspläne, ob es sich um ein sensibles Tragwerk handelt oder nicht. Wenn mögliche Fehlerquellen in diesem Bereich entstehen können, wird eine Sonderprüfung veranlasst. Diese Sonderprüfung wird meist von erfahrenen Fachleuten zu diesem Themengebiet durchgeführt.

Beim Bewertungsprozess werden folgende Kriterien herangezogen:

- Die Ergebnisse der Hauptprüfung - darunter fallen die eingetragenen Schäden und Mängel in den Schadenslisten.
- Wenn eine Sonderprüfung veranlasst wurde, dann wird auch dieses Ergebnis in die Bewertung einfließen.
- Durch die vor Ort erhoben Daten können Nachberechnungen erstellt werden, um den Zustand des Tragwerks zu errechnen. Diese Nachberechnungen werden durchgeführt, wenn z.B.: der Verdacht auf grobe Planungsfehler besteht. Wenn eine Nachberechnung veranlasst wurde, wird auch diese in die Hauptbewertung einfließen.
- Auch allgemeine Aspekte wie Alter und Streckenauslastung der Brücke sind Bewertungskriterien.

Die Entscheidungskraft der Bewertung liegt bei der Leiterin oder beim Leiter der Inspektion. Die Entscheidungsperson organisiert Veranlassungen und Mängelbehebungen. Zusätzlich ist diese Person für den Abschlussbericht, der den Revisionsblättern zugeordnet wird, verantwortlich.

Die Inspektionsleiterin oder der Inspektionsleiter muss beurteilen, ob die Brücke funktionstüchtig ist oder nicht. Tragsicherheit, Dauerhaftigkeit und Verkehrssicherheit wurden in den Richtlinien der Österreichischen Bundesbahnen für eine volle Funktionstüchtigkeit definiert. Wenn eine dieser Eigenschaften erhebliche Mängel aufweist, wird die Brücke als nicht funktionstüchtig bewertet. Dann darf die Brücke nicht mehr im alltäglichen Gebrauch im Einsatz sein bzw. verwendet werden.

4.2.2.1.4 Allgemein – Nachbearbeitung

Nachdem die Inspektion abgeschlossen und die Brücke bewertet worden ist, müssen noch ein paar Nachbearbeitungsarbeiten durchgeführt werden. Dazu zählen Veranlassungen, dass die Inspektorin oder der Inspektor bei Schäden oder

Mängel, die ein Risiko für die Funktionstüchtigkeit der Brücke darstellen. Bei Veranlassungen muss der zuständigen Behörde mitgeteilt werden, dass die Schäden und Mängel beim betreffenden Bauobjekt behoben werden müssen.

Ein weiteres Aufgabengebiet in diesem Teilprozess ist es, welche Eigenschaften und Funktionen weitere Prüfungsintervalle aufweisen. Die Prüfungsintervalle werden in drei Kategorien aufgeteilt, **laufende Überwachung** und **periodische Kontrollen** und **Hauptüberprüfungen**. Die Leiterin oder der Leiter der Inspektion ist laut Dienstvorschrift dazu verpflichtet, diese Prüfungsintervalle zu definieren.

Die laufende Überwachung kontrolliert auf Verkehrssicherheit und grobe Schäden an der Konstruktion oder der Geländeanbindung (Österreichische Bundesbahnen, 1996). Das Augenmerk bei laufenden Überwachungen sollte auf Elemente wie grobe Risse, Absenkungen, Wasseraustritte und sonstige Bauteile der Brücke gelegt werden. Wenn eine laufende Überwachung angeordnet wurde, müssen diese Tätigkeiten in Zeitabständen von 4 Monaten durchgeführt werden.

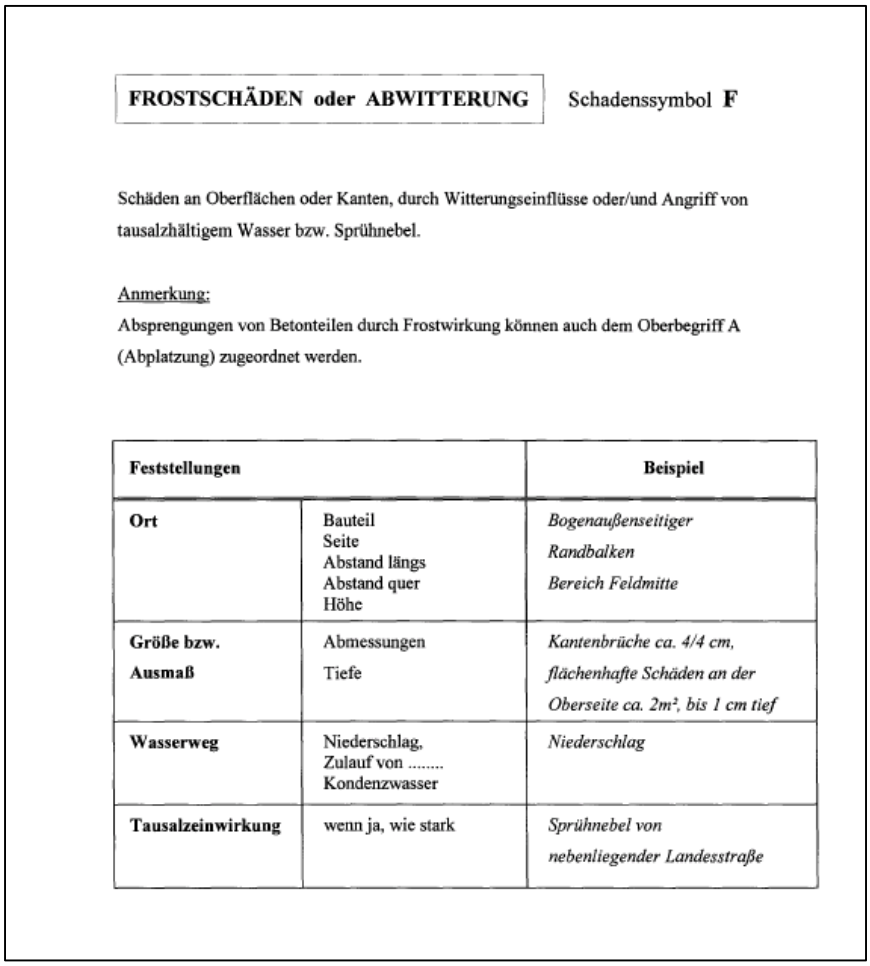
Die zweite Kategorie sind die periodischen Kontrollen: diese hat andere Charakteristiken, wie die laufende Überprüfung. Die Ausführende oder der Ausführende einer periodischen Kontrolle, muss die Aktivität der Kontrolle auf Eigenschaften wie Wasserableitungen vom Tragwerk, deutliche Risse, Korrosionsstellen oder wie Schäden an Fugen setzen. Werden bei einer periodischen Kontrolle Schäden festgestellt, die bedeutsam sind, ist einer außerordentlichen Hauptüberprüfung bzw. eine auf Niveau der Hauptüberprüfung durchzuführende Nachbegutachtung zu veranlassen (Österreichische Bundesbahnen, 1996). Periodische Kontrollen werden in Abständen von maximal 2 Jahren durchgeführt. Es besteht die Möglichkeit, dass außerordentliche Kontrollen angeordnet werden. Das tritt bei möglichen Naturereignissen wie Hochwasser, Lawinenabgang oder einem Erdbeben in Kraft.

Die dritte Kategorie ist die Hauptüberprüfung: die Eigenschaften dieser Kategorie wurde im Punkt Inspektion vor Ort angemerkt. Hauptüberprüfungen werden durch Expertinnen oder Experten der Brückenbauabteilung der ÖBB oder durch extern beauftragte Expertinnen oder Experten durchgeführt. Die Intervalldauer bei der Hauptüberprüfung liegt bei maximal 6 Jahren. Außerordentliche Hauptüberprüfungen werden durch jedes große Naturereignis und durch eine bedenkliche Schadensmeldung aus der laufenden oder periodischen Überwachung veranlasst (Österreichische Bundesbahnen, 1996).

4.2.2.1.5 Spannbetonbrückenüberwachung Inspektionshilfen

Das ÖBB Dokument wurde in zwei Teilen untergliedert. Der Allgemeine Teil behandelt die Abläufe, Strukturen und rechtlichen Hinweise. Der zweite analysierte Teil, behandelt die Inspektionshilfen bei einer Spannbetonbrückenüberwachung. In diesem Teil der ÖBB-Richtlinien werden die typischen Schäden und Mängel

noch einmal behandelt. Diese Anhänge sind für die prüfenden Inspektorinnen und Inspektoren als Hilfsmittel gedacht. Diese Hilfsmittel wurden in Form von Listen aufgebaut, dass die zuständige Inspektorin oder der Inspektor, Punkt für Punkt von der Liste abarbeiten kann. Für jede Schadenskategorie ist in den Anhängen der Richtlinien die passende Checkliste vorhanden. In der Abbildung 16 ist ein Beispiel für eine Checkliste für die Schadenskategorie Frostschäden oder Abwitterungen.



FROSTSCHÄDEN oder ABWITTERUNG Schadenssymbol **F**

Schäden an Oberflächen oder Kanten, durch Witterungseinflüsse oder/und Angriff von tausalzhältigem Wasser bzw. Sprühnebel.

Anmerkung:
Absprengungen von Betonteilen durch Frostwirkung können auch dem Oberbegriff A (Abplatzung) zugeordnet werden.

Feststellungen		**Beispiel**
Ort	Bauteil Seite Abstand längs Abstand quer Höhe	*Bogenaußenseitiger Randbalken* *Bereich Feldmitte*
Größe bzw. Ausmaß	Abmessungen Tiefe	*Kantenbrüche ca. 4/4 cm, flächenhafte Schäden an der Oberseite ca. 2m², bis 1 cm tief*
Wasserweg	Niederschlag, Zulauf von ..... Kondenzwasser	*Niederschlag*
Tausalzeinwirkung	wenn ja, wie stark	*Sprühnebel von nebenliegender Landesstraße*

Abbildung 16: Checkliste für die Schadenskategorie F (Österreichische Bundesbahnen, 1996)

In der Spalte Beispiel wurden Beispielinhalte angegeben, dass die Expertin oder der Experte einen Überblick über die Hilfestellung bekommt.

4.2.2.2 Klassifizierung von Schadenskategorien

In der Bestimmung der Kriterien für die Dokumentenanalyse war die Klassifizierung von Schadenskategorien ein wichtiger Bestandteil. Das Ziel dieses Analysepunktes war es, zu erfahren, wie die Richtlinien für Spannbetonbrücken und die Bewertung von Schadensklassen abgebildet werden. Außerdem soll ein Überblick über die Definitionen der einzelnen Schadenskategorien dargestellt werden.

4.2.2.2.1 Schadenskategorien

In den Richtlinien für Überwachung und Inspektion von Spannbetonbrücken hat die ÖBB für Schäden und Mängel, die mit großer Häufigkeit vorkommen, eigene Schadenskategorien erstellt. Eine Schadenskategorie wird aus zwei Teilen definiert: aus Schadensymbol und Beschreibung. Folgende Schadenskategorien werden in den Richtlinien definiert:

- A – Abplatzungen: Abplatzungen sind begrenzte Abbrüche an Flächen oder auch Kantenabbrüche.
- F – Frostschäden oder Abwitterung: Schaden an Oberflächen oder Kanten, durch Witterungseinflüsse oder/und Angriff von tausalzhaltigem Wasser bzw. Sprühnebel (Österreichische Bundesbahnen, 1996).
- H – Hohlstelle: Hohlstellen werden mittels Abklopfen erkannt und sind eine Vorstufe zur Kategorie A.
- K – Korrosion: Korrosion ist eine Reaktion eines metallischen Werkstoffes mit seiner Umgebung, die eine messbare Veränderung des Werkstoffes bewirkt. Von einem Korrosionsschaden spricht man, wenn die Korrosion die Funktion eines Bauteiles oder eines ganzes Systems beeinträchtigt (Czichos & Habig, 2010). Es gibt zwei Arten von Korrosion die bei Brücken entstehen können, die Betonkorrosion und Stahlkorrosion. Für die Brückeninspektorinnen oder Brückeninspektoren werden in den Hilfsmitteln für Brückeninspektionen die möglichen Stellen vordefiniert. Beispiele dafür sind Geländer, Spannstahl, Anker oder Mastverankerung.
- N – Nest: Nester sind Bereiche, die eine offene Struktur oder einen Hohlraum besitzen.
- P – Poröser Beton: Poröser Beton ist ein Mangel der Betonsichtflächen. Lunker und grobe Poren verhindern eine geschlossene, mit Zementschlamme gedeckte Oberflache (Österreichische Bundesbahnen, 1996).
- R – Riss: Es müssen alle Risse dokumentiert, die quer zur vorgespannten Haupttragrichtung vorhanden sind, auch in sensiblen Bereichen müssen auch sehr feine Risse dokumentiert werden. Die Ursachen von Rissen sind sehr unterschiedlich, damit die Inspektorinnen und Inspektoren einen Überblick über die Art des Risses und die mögliche Ursache haben, wurde in den Richtlinien eine Tabelle definiert. Die Abbildung 17 ist ein Auszug von dieser Tabelle.

Rißart	mögliche Ursache
Risse in der Zugzone, quer zur Tragrichtung. In Stegen meist durchgehend und auf beiden Seiten sichtbar. a) Stahlbeton, feine Risse b) Risse bei Spannbeton c) wenige, grobe Risse	Lasteinwirkung oder Zwang a) normale Zustand II-Risse b) Zwänge, Überlastung, Vorspanndefizit c) zu wenig bzw. schlecht verteilte, zur Rißweitenbegrenzung ungeeignete Bewehrung
Schräge Risse in Auflagernähe a) sehr fein b) starke Risse	Wirkung von Querkraft (u. Torsion) a) akzeptabel b) Schubbewehrungsdefizit, Zwänge
Risse in neutralen Zonen	Zwänge - Stützensenkung - Temperaturgradient u.a.
Diffuse Oberflächenrisse, haarfein	Zwänge aus Frühschwinden, Abfließen der Abbindewärme .....
Risse in Tragrichtung, lotrechte Risse in WL-Mauern und Wänden	Schwindspannungen
Schrägrisse bei Lasteinleitungen (Lager, Spannköpfe)	Spaltzug- und Randzugkräfte
Betonoberseite, seichte Risse, dem Bewehrungsnetz folgend	Absetzrisse, entstanden in der ersten Abbindephase

Abbildung 17: Der erste Teil der Hilfstabelle der möglichen Rissarten (Österreichische Bundesbahnen, 1996)

- Ü – Überdeckungsmangel: In der Fachsprache bei Brückeninspektorinnen und -inspektoren wird von Überdeckungsmangel gesprochen, wenn eine zu geringe Betondecke vorhanden ist. Dadurch können Folgeschäden wie Korrosion entstehen.
- V – Verpressfehler: Von einem sogenannten Verpressfehler spricht man, wenn Hüllrohre verpresst sind und vom Zement freiliegt. Auch von einem Verpressungsfehler wird gesprochen, wenn das Hüllrohr nur teilweise gefüllt ist oder wenn im Beton Hohlräume gefunden werden. In der Abbildung 18 wird im Symbol 5 beschrieben, welcher Bauteil genau ein Hüllrohr ist.

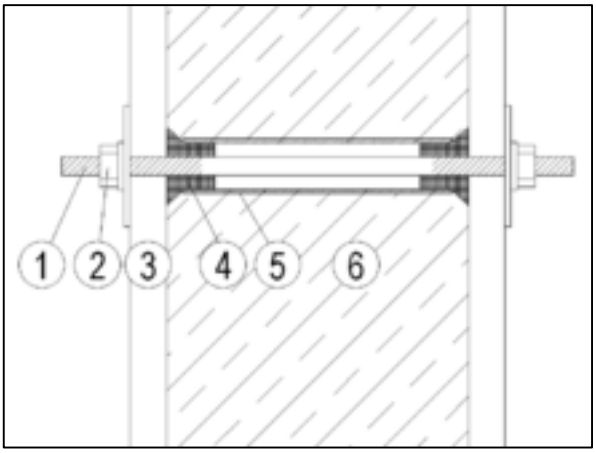


Abbildung 18: Hüllrohr ist das Symbol 5 in der Symbolgrafik („Schalungsanker“, 2017)

- W – Wasseraustritt - Nässe: Wasseraustritt und Nässeprobleme sind meistens Folgereaktionen auf nachstehende Mängel. Praxisrelevante Beispiele dazu wären undichte Fahrbahnübergänge, undichte oder verlegte Entwässerungsrohre oder undichte Fugen.

Diese zehn Schadenskategorien sind die Vorlage bzw. das Muster, das bei der Schadenssuche die größte Relevanz für die Inspektorinnen und Inspektoren ist. Zusätzlich wird das Fachpersonal durch Inspektionshilfen im Bereich der Schadenskategorie unterstützt.

4.2.2.2.2 Schadensklassen

Zusätzlich zu den Schadenskategorien, die mit dem Schadensymbol und Beschreibung definiert sind, gibt es eine weitere Klassifizierung. Die Rede ist von der Schadensklassifizierung: um Fehler und Mängel noch eindeutiger zuordnen zu können, wurde ein zusätzlicher Parameter installiert. Die Schadensklassen sind in folgende Klassen aufgeteilt:

- Klasse 1: Keine oder geringe Schäden oder optische Mängel. Schäden in dieser Klasse müssen sich keiner Sanierung unterziehen, auch der Dokumentation müssen diese Schäden und Mängel nicht aufgezeigt werden.
- Klasse 2: Leichte und begrenzte Mängel und Schäden. In der Definition haben diese Schäden keine Beeinträchtigung der Funktionalität der Brücke, wenn diese Schäden nicht sofort behoben werden.
- Klasse 3: Mittlere Schäden oder Schäden großen Umfanges, die aus Gründen der Erhaltung der Bausubstanz mittelfristig (bis zur nächsten Brückenprüfung) saniert werden sollten (Österreichische Bundesbahnen, 1996). Beispiele dafür sind Betonmängel und Korrosionsschäden.

- Klasse 4: Schwere Schäden oder Schäden größeren Umfanges, die Einfluss auf die Tragfähigkeit des Bauwerkes haben können und daher mittelfristig saniert werden müssen oder Sofortmaßnahmen erfordern (Österreichische Bundesbahnen, 1996). Beispiele für diese Schadensklasse sind starke Korrosionen auf Lager oder Spannkabel.

Zusätzlich können die Schadensklassen als Instrument für Bewertungsmöglichkeiten agieren. In dieser Form der Bewertung werden alle möglichen Bauelemente einer Brücke aufgelistet und mit der jeweiligen Schadensklasse bewertet. In der Abbildung 19 ist ein Beispiel, wie eine Bewertung anhand einer Schadenstabelle aussieht.

ZUSTANDSBEWERTUNG
MIT SCHADENSTABELLE
(Beispiel)

Bauteil \ Schadensklasse		1	2	3	4
Geländeanbindung Uferböschung		X			
Unterbau:	WL Astadt		X		
	WL Bemarkt		X ÷	X	
Brückenlager				X	
Tragwerk	Hauptträger		X		
	Querträger	X			
	Fahrbahnplatte		X ÷	X	
Ausrüstung	Abdichtung			X	
	Entwässerung		X		
	Randleisten und Geländer			X	
	Fahrbahnübergang		X		
Gleisanlage	Schotterbett	X			
	Gleis		X		

Abbildung 19: Beispiel Zustandsbewertung anhand einer Schadenstabelle (Österreichische Bundesbahnen, 1996)

4.2.2.3 Zusammenfassung

Die ÖBB-Richtlinie für die Überwachung und Inspektion von Eisenbahnbrücken in Spannbetonbauweise ist ein erheblicher Faktor für die Anforderungserhebung und für die Entwicklung des Konzeptes. Hauptfaktor für die Priorität dieses Dokumentes ist es, dass dieses als Basiswissen für die Durchführung von Brückeninspektionen notwendig ist.

Relevante Faktoren für das Konzept sind, wie der genaue Ablauf einer Inspektion definiert ist. Welche Normen verwendet werden und welches Personal die Inspektionen durchführen dürfen. Weitere Kriterien für das Konzept ist die Vorbereitung und Nachbereitungsphase einer Inspektion. Es muss für das Konzept und in weiterer Folge für den Prototyp eine Systematik definiert werden, um Datensätze zu früheren und späteren Zeitpunkten noch verändern zu können.

Zusätzlich müssen Maßnahmen definiert werden, um die Sicherheit für Brücken in der Zukunft zu gewährleisten. Es müssen Interpretationen bestimmt werden, die eine Veranlassung oder andere Maßnahmen, die zur Erhaltung der Brücken dienen, zuverlässig ausführt. Aufgrund dieser Bestimmungen müssen die zuständigen Behörden Mitteilungen senden, damit die Erhaltungsmaßnahmen durchgeführt werden.

Ein weiterer wichtiger Prozess ist die Bewertung von Schäden und Mängeln. In den Richtlinien hat die ÖBB ein Modell definiert, das aus Mustervorlagen die Schadenskategorie auswählt. Es gibt zehn verschiedene Kategorien und ein Schaden oder Mangel wird zu dieser Kategorie zugewiesen. Weiters bietet dieses Modell eine zusätzliche Integration der Bewertung von Schäden. Die Schadensklasse bewertet in eine Skala von 1 bis 4 Teile und Objekte einer Brücke.

4.2.3 Dokument Bundesministerium für Verkehr, Bau und Stadtentwicklung: Bauwerksprüfung nach DIN 1076 Bedeutung, Organisation, Kosten

Diese Dokumentation wurde im Jahr 2013 vom deutschen Bundesministerium für Verkehr, Bau und Stadtentwicklung veröffentlicht. Die vorliegende Dokumentation begründet die rechtliche, technische und wirtschaftliche Notwendigkeit der Prüfung von Ingenieurbauwerken und gibt einen Überblick über die Organisation, Durchführung und Kosten der Bauwerksprüfung(Bundesministerium für Verkehr, Bau und Stadtentwicklung, 2013).

Das Dokument behandelt alle Bereiche, die in der Straßenbauverwaltung eine gewisse Relevanz besitzen. In diesen Prüfungsprozess fallen dann auch die Brückenüberprüfungen hinein. Die Standards und Norm dieser Richtlinien wurden allgemein formuliert um sie auf andere Bauobjekte wie Tunnel oder sonstige Straßenbauobjekte anzuwenden.

4.2.3.1 Gliederung und Aufteilung der Teilprozesse

4.2.3.1.1 Vorbereitung und Durchführung

Der Überprüfungsprozess wurde in verschiedene Phasen gegliedert um den Qualitätsprozess ordnungsgemäß durchzuführen. Der erste Bereich ist die **Vorbereitung**.

In diesem Teilprozess werden die Maßnahmen und Richtlinien, die für eine Inspektion benötigt werden, definiert. Große Unterschiede zu den Richtlinien der ÖBB sind nicht vorhanden. Folgende Instrumente und Tätigkeiten sind in der Vorbereitung zu beachten:

- Rechtzeitige Planung und Organisation von benötigten Instrumenten wie Prüfgeräte und Besichtigungsgeräte.
- Organisation von Verkehrssicherheitsmaßnahmen, es müssen die nötigen Genehmigungen eingeholt werden, damit eine Inspektion durchgeführt werden darf.
- Bereitstellung eines Inspektionsteams und die Abstimmung eines geeigneten Termins.
- Durchführung von Maßnahmen, die vor Ort durchgeführt werden - wie den Zugang zum Objekt freimachen.

Die nächste Phase ist die **Durchführung** der Inspektion und dazu müssen folgende Leistung erbracht werden:

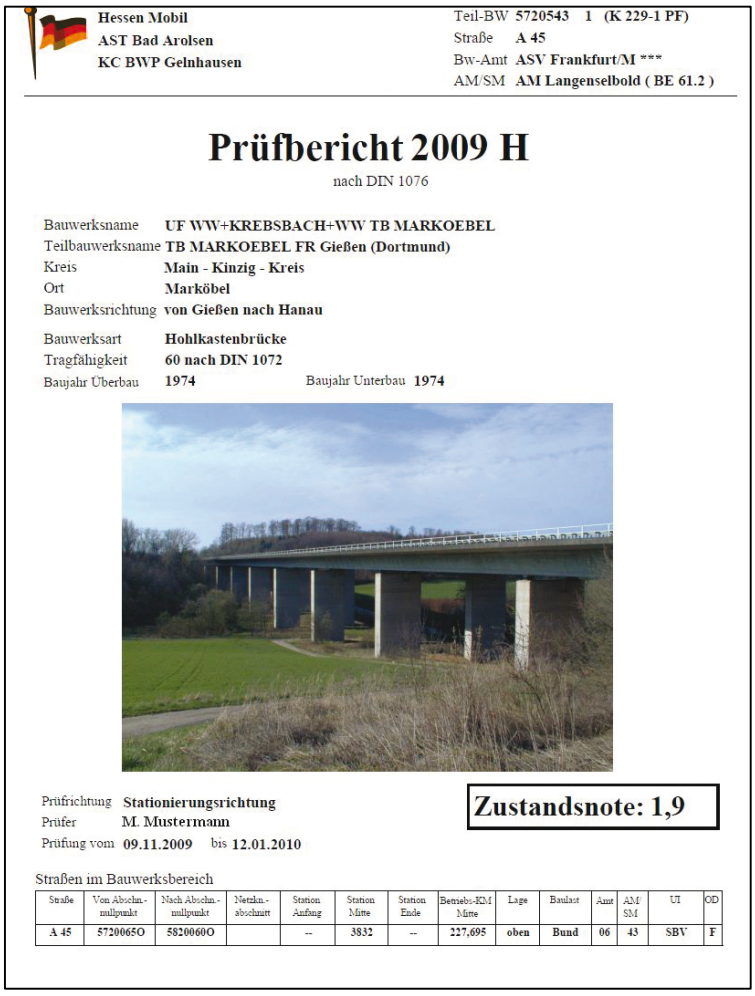
- Die Arbeitsstelle absichern und gewährleisten, dass die höchste Sicherheit für das Inspektionsteam besteht.
- Koordinierung und Arbeitszuteilung für die Inspektorinnen und Inspektoren.
- Durchführung der Inspektion - zusätzlich erfolgt die Dokumentation von Schäden und Mängel.

4.2.3.1.2 Dokumentation und Bewertung

Die Dokumentation erfolgt in einem Prüfbericht. In diesem Prüfbericht müssen alle Messdaten, Ergebnisse einer möglichen Zusatzuntersuchung, Skizzen und Fotos der Inspektion enthalten sein. Die Erfassung von Schäden und Mängel wird mit dem System SIB-Bauwerke aufgenommen. Diese Applikation wurde im Punkt Forschungsprojekte näher beschrieben.

Zusätzlich gibt es einen Schadenskatalog der 1700 Schadensbeispiele enthält, der in SIB-Bauwerke integriert ist. Zu jedem Schaden oder Mangel muss dazu der passende Schaden aus dem Katalog ausgewählt werden. Die Fotodokumentation und die dazu gemachten Skizzen werden auch in die Software integriert, um den Schadensbericht ausführlicher zu verfassen. Die Software verfügt über eine Möglichkeit, die erforderlichen Maßnahmen für eine Instandhaltung zu generieren.

Diese Auswertung von Maßnahmen wird als Empfehlung definiert. Anhand der eingegebenen Schäden mit ihrer Schadensklasse, ermittelt die Applikation automatisch eine Zustandsnote für das Bauobjekt. In der Abbildung 20 ist ein Beispielbild eines Prüfberichtes. Wie zu erkennen ist, wird die Zustandsnote gleich sichtbar am Deckblatt deklariert.



Hessen Mobil
AST Bad Arolsen
KC BWP Gelnhausen

Teil-BW 5720543 1 (K 229-1 PF)
Straße A 45
Bw-Amt ASV Frankfurt/M ***
AM/SM AM Langenselbold (BE 61.2)

Prüfbericht 2009 H

nach DIN 1076

Bauwerksname UF WW+KREBSBACH+WW TB MARKOEBEL
Teilbauwerksname TB MARKOEBEL FR Gießen (Dortmund)
Kreis Main - Kinzig - Kreis
Ort Marköbel
Bauwerksrichtung von Gießen nach Hanau

Bauwerksart Hohlkastenbrücke
Tragfähigkeit 60 nach DIN 1072
Baujahr Überbau 1974 Baujahr Unterbau 1974

Prüfrichtung Stationierungsrichtung
Prüfer M. Mustermann
Prüfung vom 09.11.2009 bis 12.01.2010

Zustandsnote: 1,9

Straßen im Bauwerksbereich

Straße	Von Abschn.-nullpunkt	Nach Abschn.-nullpunkt	Netzkn.-abschnitt	Station Anfang	Station Mitte	Station Ende	Betriebs-KM Mitte	Lage	Baulast	Amt	AM/SM	UI	OD
A 45	5720065O	5820060O		--	3832	--	227,695	oben	Bund	06	43	SBV	F

Abbildung 20: Beispielbild eines Prüfberichtes (Bundesministerium für Verkehr, Bau und Stadtentwicklung, 2013)

Bereits im Verlauf und nach Abschluss der Bauwerksprüfung werden durch die Ingenieurin oder den Ingenieur der Bauwerksprüfung mögliche Schadensursachen ergründet und in kritischen Fällen umgehend geeignete Maßnahmen veranlasst, die die Sicherheit der Verkehrsteilnehmer und Dritter gewährleisten (Bundesministerium für Verkehr, Bau und Stadtentwicklung, 2013). Die Maßnahmen zur Erhaltung der Brücke werden an die zuständige Dienststelle weitergegeben.

4.2.3.2 Klassifizierung von Schadenskategorien

In der direkten Gegenüberstellung der Dokumentenanalyse mit den Richtlinien der ÖBB werden Schadenskategorien nicht so definiert. Ein möglicher Grund wäre eine andere Definition von Kategorien. Die Applikation, die bei Brückeninspektionen im Einsatz ist, hat bereits die Schadenskategorien inklusive genauer Schadensbeschreibung vordefiniert. Im Katalog sind bis zu 1700 mögliche Schäden und Mängel bereits vordefiniert. Die Inspektorin oder der Inspektor muss dadurch mit der Software geübter sein, um den Schaden oder Mangel gleich der richtigen Kategorie zuzuweisen.

Die vordefinierten Muster handhaben die Standards und Normen, die bereits vom Bundesministerium für Verkehr, Bau und Stadtentwicklung vorgegeben wurden.

4.2.3.3 Zusammenfassung

Die einzelnen Teilprozesse und Methoden weisen in den meisten Bereichen keinen markanten Unterschied zur Dokumentenanalyse von Richtlinie für die Überwachung und Inspektion von Eisenbahnbrücken in Spannbetonweise, auf. Unterschiede gibt es in den Bereichen der Dokumentation und dann als Folgereaktion bei der Schadensklassifizierung. Da beim Inspektionsprozess bereits mit technischen Entwicklungsoptionen gearbeitet wird, hat das deutsche Bundesministerium bereits einen innovativeren Ansatz im Bereich der Verwaltung. Es wurde mit Datenbanken und den zur Verfügung gestellten Schnittstellen eine zentralisierte Plattform für die Verwaltung von Bauobjekten erstellt.

Diese Möglichkeiten bringen eine große Flexibilität und Erweiterungsmöglichkeiten mit. Zusätzlich werden durch Features, wie den Prozess der automatisierten Generierung von möglichen Maßnahmen zur Instandhaltung von Brücken, den Inspektorinnen und Inspektoren Hilfestellungen empfohlen.

4.2.4 Dokument ÖBB: Revisionsblatt

4.2.4.1 Analyse

Das Revisionsblatt ist ein Teildokument und dient zur Verwaltung von Inspektionsdaten bei Brückenuntersuchungen. Nach einer abgeschlossenen Brückeninspektion, wird das Revisionsblatt von der zuständigen Inspektionsleiterin oder dem Inspektionsleiter archiviert.

In der Analyse werden die einzelnen Eigenschaften eines Revisionsblattes analysiert und erklärt. Vorweg, das analysierte Dokument ist ein Originaldokument, das bei Inspektionen verwendet wurde. Aus Datenschutzgründen wurden manche Textteile unlesbar gemacht.

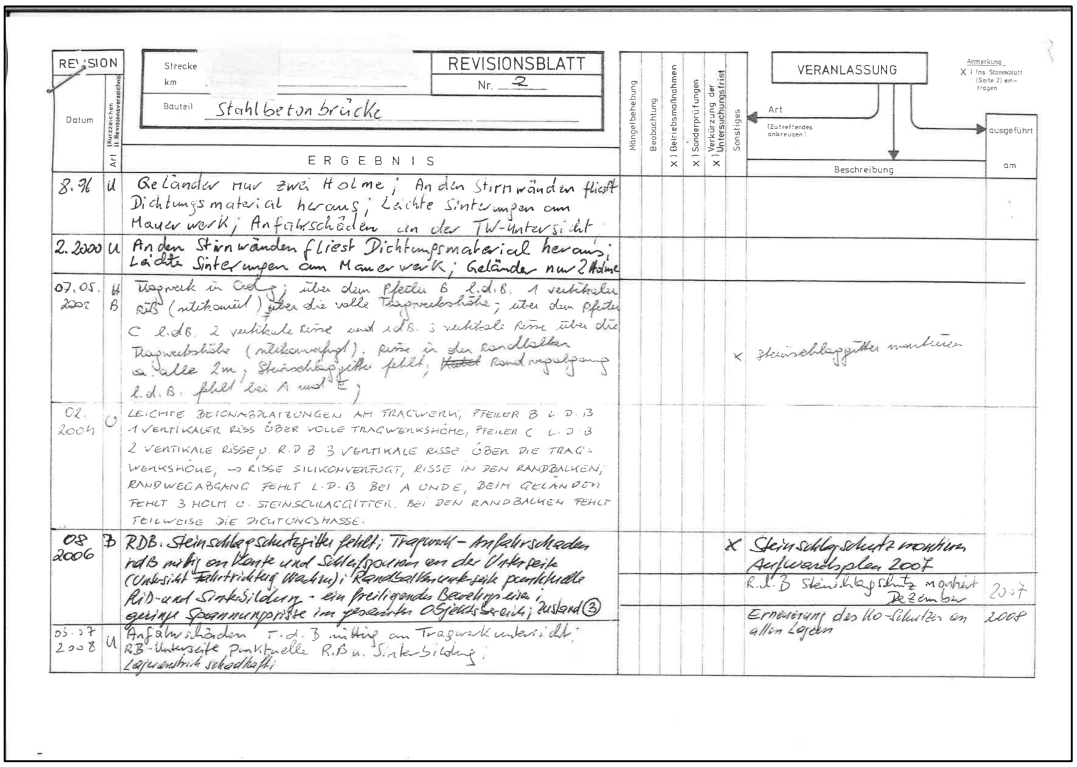


Abbildung 21: Revisionsblatt bei Brückeninspektionen

Der erste Teil der beschrieben wird, sind die allgemeinen Definitionen zur Brücke bzw. zum Bauobjekt.

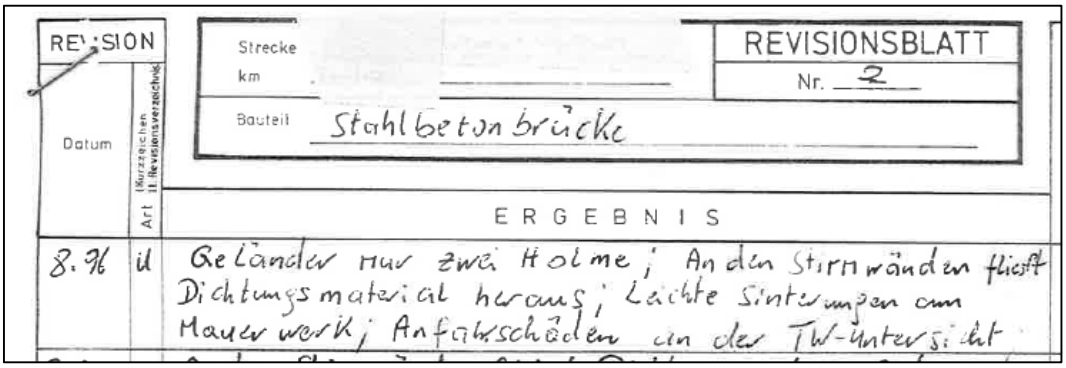


Abbildung 22: Revisionsblatt Brückeninformationen

Damit die Inspektion der richtigen Brücke zugewiesen werden kann, müssen allgemeine Information zur Brücke gegeben sein. Die Parameter dazu sind Strecke, km-Stelle an der sich das Bauobjekt befindet und das zu untersuchende Bauteil. Wie auf der Abbildung zu erkennen ist, wird in der Spalte Datum, das Inspektionsdatum eingetragen, wann die Inspektion durchgeführt wurde. In diesem Dokument werden nur Monat und Jahr angegeben. Die Resultate der Inspektion werden in der Spalte Ergebnis notiert.

Die nächste Eigenschaft des Dokumentes ist die Veranlassung und ihre Funktionalität. Die Abbildung 23 ist ein vergrößerter Auszug des Originaldokuments.

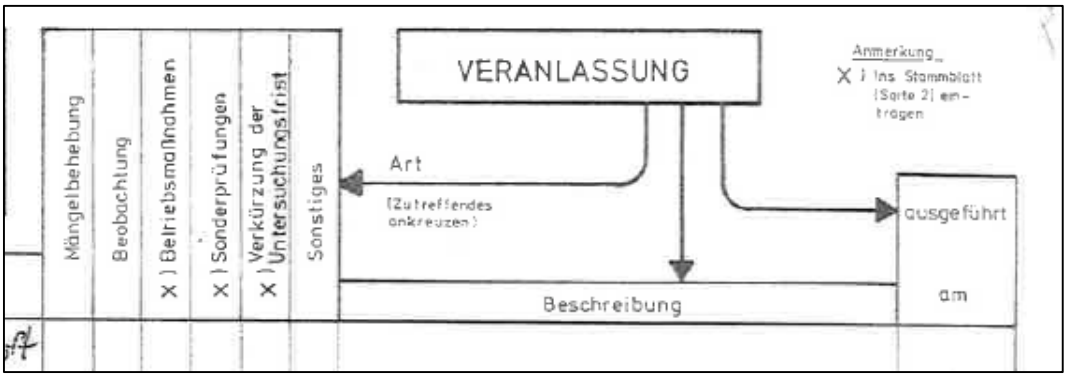


Abbildung 23: Revisionsblatt Veranlassung

Wie im Inspektionsprozess definiert, müssen bei groben Mängeln und Schäden Veranlassungen getroffen werden. Diese Veranlassungen müssen durchgeführt werden, um die Sicherheit und die Funktionalität der Brücke zu gewährleisten.

Die Veranlassung besteht aus drei Parametern: der erste Parameter ist die **Art -** hier wird definiert, aus welchem Grund eine Veranlassung durchgeführt wird. Hier sind die folgenden Auswahlmöglichkeiten:

- Mängelbehebung
- Beobachtung
- Betriebsmaßnahmen
- Sonderprüfungen
- Verkürzung der Untersuchungsfirst
- Sonstiges

Der zweite Parameter ist die **Beschreibung** - hier wird definiert, warum eine Veranlassung durchgeführt wird. Der letzte und dritte Parameter ist das **Ausführungsdatum** – dieser gibt an, bis zu welchem Datum die Veranlassung durchgeführt werden muss.

In der Abbildung 24 werden Veranlassungen getroffen, weil der Steinschlagschutz fehlte. So eine Veranlassung fällt in die Kategorie Sonstiges.

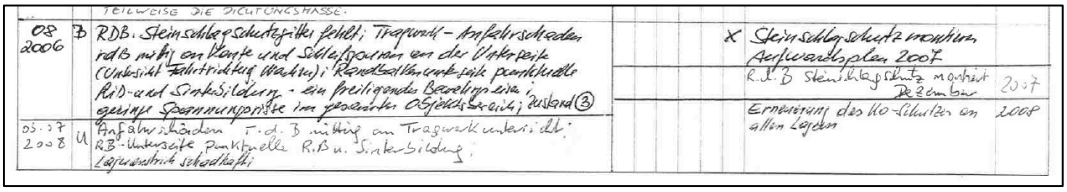


Abbildung 24: Beispiel für eine Veranlassung „Sonstiges“

Wie im Beispiel zu erkennen ist, wurde das Veranlassungsdatum für 2007 gesetzt.

Eine weitere Thematik im Revisionsblatt ist die Handhabung von Fotomaterial. Die folgende Grafik ist ein Beispiel, wie die Verwaltung von Fotomaterial aussieht. Material wie in der Abbildung 25 wird an das Revisionsblatt angefügt.

Abbildung 25: Fotomaterial einer Brückeninspektion

4.2.4.2 Problemstellungen

Wie bei den Originaldokumenten zu erkennen ist, wurden die Inspektionsergebnisse handschriftlich verfasst. Das stellt eine gewisse Problematik dar, denn manche Schreibstile sind nicht gut lesbar und daher wird die Arbeit für nachfolgende Inspektorinnen oder Inspektoren erschwert.

Weiters werden in der Praxis keine Schadenskategorien oder Schadensklassen im Revisionsblatt angeben. Mögliche Gründe könnten sein, dass die Schäden nicht eindeutig zuzuordnen sind. Ein anderer Grund wäre es, dass in der Praxis die Beschreibung eine höhere Priorität hat, als eine Schadensklasse.

Eine weitere Problemstelle ist das Fotomaterial - wenn Skizzen oder Fotomaterial zur Inspektion hinzugefügt werden, müssen sie im Ergebnis auf dem Anhang vermerkt werden.

4.3 Zusammenfassung

Die Problemanalyse behandelte zwei Teile: der erste Teil war das Interview mit der Expertin im Bereich Bahntechnologie & Mobilität und der zweite Teil war die Dokumentenanalyse. Diese zwei Aspekte wurden ausgewählt, um den Prozess der Brückeninspektion aus verschieden Blickwinkeln zu betrachten. Zusätzlich wurden manche Teilprozesse sehr genau analysiert um mögliche Problemstellungen zu finden.

Um keine Voreingenommenheit und Abhängigkeit von ÖBB Dokumenten zu schaffen, wurde auch ein Dokument von einer anderen Institution begutachtet und analysiert. Daraus wurden weitere Informationen und Problemstellungen im Bereich der Brückeninspektion gesammelt. Viele Erkenntnisse aus den Anforderungserhebungen überschneiden sich und bestätigen damit, dass mögliche Schwierigkeiten im Prozess verankert sind.

Die Problemanalyse ist das Grundkonzept um das Anforderungsprofil im Punkt Anforderungen und technische Entscheidungen zu definieren.

5 Anforderungen und technische Entscheidungen

Die Entwicklung des Konzeptes besteht aus zwei großen Hauptpunkten. Der erste Punkt ist das **Anforderungsprofil** - dieses wird aus den Erkenntnissen der Problemanalyse erstellt. Der zweite Hauptpunkt ist die **technische Umsetzung** - welche mobile Entwicklungsoption bei der Umsetzung des Prototyps verwendet wird.

5.1 Anforderungsprofil

In diesem Punkt werden die Erkenntnisse aus der Problemanalyse zusammengefasst. Es fließen sowohl das Expertinnen Interview, als auch die Dokumentenanalyse ein. Ziel ist es, die Kriterien, Richtlinien und Verbesserungsvorschläge in das Anforderungsprofil einfließen zu lassen. Die Anforderungen müssen so definiert werden, damit sie später als Funktionalität im Prototyp umgesetzt werden können. Bei jedem Anforderungspunkt sollen folgende Kriterien erfüllt werden:

- Kann die Funktionalität mit mobilen Entwicklungstechnologien umgesetzt werden?
- Kann man den derzeitigen Teilprozess mit diesem Anforderungspunkt verbessern?

Außerdem wird noch beschrieben, wie der aktuelle Prozess in diesem Anforderungspunkt aussieht.

5.1.1 Allgemeine Anforderungen

Das konkrete Hauptziel der kompletten Anforderungspalette ist die Entwicklung einer Applikation. Die genauere Spezifikation über die Funktionalitäten kommt in den nachfolgenden Punkten. Im Inspektionsprozess werden mobile Geräte wie Tablets und Smartphones mitgeführt. Anforderung ist es, dass eine Applikation auf diesen Zielgeräten läuft, mit dem der Inspektionsprozess unterstützt werden kann. Im zweiten Teil des Konzeptes, wird die mobile Entwicklungsoption beschrieben, mit der die Umsetzung stattfindet.

Eine weitere allgemeine Anforderung wurde aufgrund des Expertinnen Interviews mit hoher Priorität definiert. Inspektorinnen und Inspektoren sollen die Möglichkeit bekommen, auch vom Desktop auf die Applikation zuzugreifen. Grund dafür ist die

Nachbearbeitung und mögliche Korrektur von Inspektionsdaten. Diese Prozesse werden oft im Büro auf Desktopgeräten ausgeführt.

5.1.2 Brückenverwaltung

Ein Kernpunkt ist die Verwaltung von Ressourcen insbesondere Brücken. Derzeit werden die Unterlagen von Brücken in Papierform abgelegt. Ziel soll es sein, dass die Inspektorin oder der Inspektor ein Managementtool für die Verwaltung von Brücken hat. Die Funktionalität sollte so abgebildet sein, dass die Userin oder der User der Applikation, Brücken erstellen können. Zusätzlich sollte auch die Möglichkeit bestehen, Brücken zu bearbeiten, falls Fehler beim Erstellprozess gemacht wurden. Weiteres sollte bei einer großen Menge von Dateneinträgen eine Suchmaske vorhanden sein, um den richtigen Datensatz für die Brückeninspektion rasch finden zu können. Die Definition einer Brücke wird vom Revisionsblatt abgeleitet und wird folgendermaßen abgebildet:

- Strecke: Auf welcher Strecke befindet sich das Bauobjekt.
- km-Stelle: Genaue Angabe der Kilometer Stelle der Brücke.
- Bauteil: Welches Bauteil bzw. aus welchem Baumaterial besteht die Brücke.

5.1.3 Inspektionsverwaltung

Der nächste Punkt ist die Verwaltung von Inspektionen - der aktuelle Prozess dazu sieht so aus: Die Inspektionen werden am Revisionsblatt chronologisch nach dem Ausführen der Inspektion angeordnet. In der Applikation müssen die Inspektionen so abgebildet werden, dass die zuständige Bearbeiterin oder der Bearbeiter sofort einen Überblick über die Historie der Inspektionen hat. Die Parameter wie eine Inspektion aussieht, werden auch vom Revisionsblatt abgeleitet. Eine Inspektion wird mit folgenden Eigenschaften definiert:

- Datum: Wann wird die Inspektion ausgeführt?
- Ergebnis: Wie sieht das Ergebnis der Inspektion aus? Welchen Schäden und Mängel sind der Inspektorin oder dem Inspektor bei der Brückeninspektion aufgefallen?

Außerdem soll die Möglichkeit bestehen, auch Maßnahmen wie Sonderüberprüfungen bei groben Mängel oder Schäden hinzuweisen. Deswegen müssen auch die Parameter der Veranlassung zur Inspektion gegeben sein.

- Art der Veranlassung: Es gibt verschiedene Arten von Veranlassungen und folgende Möglichkeiten werden vordefiniert: Mängelbehebung, Beobachtung, Betriebsmaßnahmen, Sonderprüfungen, Verkürzung der Untersuchungsfrist und Sonstiges.

- Beschreibung: Hier wird beschrieben, aus welchen Gründen eine Veranlassung getroffen wird und welche Mängel die zuständige Abteilung beheben muss.
- Ausführungsdatum: Bis zu welchem Tag die Veranlassung durchgeführt werden muss. Das Veranlassungsdatum steht in einer Abhängigkeit mit der Art der Veranlassung.

Bei der Verwaltung von Inspektionen muss die Funktionalität gegeben sein, dass die Benutzerin oder der Benutzer der Applikation, Inspektionen erstellen kann. Inspektionen werden genau nach dem Schema, der oben genannten Eigenschaften erstellt.

Zusätzlicher Punkt dabei ist die Abhängigkeit einzelner Eigenschaften. Wann Veranlassungen erstellt werden, muss bei jeder Art der Veranlassung auch ein Ausführungsdatum bestimmt werden. Dieses Kriterium beugt Dateninkonsistenz vor und verhindert mögliche Versäumnisse bei Instandhaltungsmaßnahmen.

Weitere Anforderungen für den Bereich der Inspektionsverwaltung sollten die Editierbarkeit von Dateneinträgen sein. Auch die Möglichkeit, dass Inspektionseinträge gelöscht werden können, muss bestehen.

5.1.4 Dokumentationsmöglichkeiten

Um eine vollständige und genaue Dokumentation für die Brückeninspektion durchzuführen, muss die Applikation auch die Verwaltung von Bildmaterial im Anforderungsprofil haben. Die Dokumentation mit Fotomaterial ist einer der Kernpunkte im Prozess der Dokumentation. Die aktuelle Handhabung ist, dass die Fotos an das Revisionsblatt angehängt werden. Auch im Expertinnen Interview war das ein wichtiges Thema für die Bestimmung von Verbesserungsvorschlägen. Ziel ist es, dass Inspektorinnen und Inspektoren die Möglichkeit bekommen, Fotomaterial von älteren Inspektionen begutachten zu können. Die Funktionalität soll als Hilfsmittel für einen sauberen Dokumentationsweg dienen.

Die Applikation sollte auch die Möglichkeit besitzen, mehrere Bilder einer Inspektion zuweisen zu können. Anwenderinnen und Anwender sollen Bilder editieren und löschen können, falls unbrauchbares Material zugewiesen wurde. Außerdem soll die Opportunität gegeben sein, dass Inspektorinnen und Inspektoren nachträglich Fotomaterial vom Desktop oder vom mobilen Gerät der Inspektion hinzufügen können.

Eine weitere Möglichkeit um die Dokumentation von Schäden und Mängel besser darzustellen, ist die Skizzierung. Wie in den Richtlinien der ÖBB für den Inspektionsprozess angegeben wurde, ist die Form von Skizzen ein bewährtes Mittel um Schäden und Mängel verständlicher darzustellen. Die Applikation sollte

die Fähigkeit besitzen, Skizzen darzustellen. Alternative Möglichkeit um diese Funktion abzubilden, wäre das Abfotografieren von Skizzen.

5.1.5 Dokumentation von Schäden

Die Dokumentation von Schäden sollte allgemeiner definiert werden. Grund dafür ist der Einsatz von verschiedenen Standards und Normen. Die Applikation sollte so interpretiert werden, damit auch andere Standards und Normen von Schadensklassen und Schadenskategorien eingesetzt werden können. Die Definition eines Schadens wird in der Applikation mit folgenden Parametern gekennzeichnet:

- Schadensbeschreibung: Hier wird beschrieben, um welchen Schaden es sich handelt.
- Schadensklassifizierung: Allgemeine Definition von Schadensklassen je nach dem Standard oder Norm nach dem die Inspektorinnen oder Inspektoren die Brücke untersuchen. Wie in der Problemstellung erkannt wurde, gibt es verschiedene Schadensklassen und Kategorien. Um die Anwenderin oder den Anwender nicht mit einem Standard oder einer Norm einzuschränken, der nicht für die untersuchende Brücke tauglich ist, wird die Schadensklassifizierung allgemein definiert.

Eine Anforderung aus dem Expertinnen Interview war das Markieren von Schäden am Fotomaterial. Die Benutzerin oder der Benutzer soll die Möglichkeit haben, auf dem Fotomaterial mögliche Schäden und Mängel zu markieren. Im Prototyp soll die Funktionalität so abgebildet werden, dass per Klick ein Marker am Fotomaterial gesetzt wird.

Jeder gesetzte Marker repräsentiert einen Schaden, das heißt jeder Marker hat also die gleichen Parameter wie ein Schaden und muss genauso mit Beschreibung und Schadensklassifizierung bestimmt werden.

5.1.6 Archivierungsverwaltung

Ein weiteres Thema im Anforderungsprofil ist die Verwaltung von Dokumenten in Papierform. Bei Behörden und anderen Institutionen werden die Brückeninspektionsdokumente in Papierform abgelegt, das heißt die Applikation muss eine Funktion liefern, um diese Anforderung abzudecken.

Im derzeitigen Prozess wird das Originaldokument aufbewahrt. In der zuständigen Behörde wird das Revisionsblatt wird im Dokumentenarchiv abgelegt. In der Applikation muss eine Funktion gegeben sein, um das Dokument zu generieren, damit die zuständige Institution das Dokument auch in Papierform archivieren kann.

Die Applikation soll eine PDF Datei generieren, die dann in den Archiven abgelegt werden kann.

Die Exportiermöglichkeiten sollen für mehrere Varianten möglich sein. Die Anwenderin oder der Anwender soll die Möglichkeit haben, alle Inspektionsdaten von einer Brücke exportieren zu können. Aber zusätzlich soll es auch möglich sein, dass die Daten eines Vorganges einer speziellen Inspektion exportiert werden können.

5.1.7 Benutzerverwaltung und Sicherheit

Ein wichtiger Punkt ist die Verwaltung von Benutzern, jede Inspektorin und jeder Inspektor sollte ein Profil anlegen können, das mit Benutzername und Passwort versehen ist. In den Inspektionsdatensätzen sollte vermerkt werden, wer die Inspektion erstellt und wer sie möglicherweise bearbeitet hat.

Wenn diese Merkmale in den Inspektionsdaten vermerkt werden, haben spätere Inspektorinnen oder Inspektoren eine Übersicht, wer die Inspektion durchgeführt hat und können bei Unklarheiten die betreffende Person kontaktieren.

Weiters gewährleistet dieses Modul auch Sicherheit, denn durch die Benutzerverwaltung kann die Applikation für nicht autorisierte Benutzerinnen und Benutzer abgekapselt werden. Dadurch können keine Daten von nicht autorisierten Personen gelöscht oder geändert werden.

5.2 Technische Umsetzung

5.2.1 Mobile Entwicklungstechnologien

Für die Wahl der mobilen Entwicklungstechnologie haben zwei Faktoren eine Rolle gespielt: der erste Faktor war das Anforderungsprofil und der zweite Faktor war die State of the Art Analyse der mobilen Entwicklungstechnologien. Diese Kernpunkte aus dem Anforderungsprofil waren entscheidende Kriterien bei der Auswahl der Methodik für die Umsetzung des Prototyps:

- **Plattformunabhängigkeit:** Die Applikation muss auf allen Betriebssystemen laufen. Zusätzliche Anforderung ist es, dass die Applikation auch am Desktop laufen soll.
- **Datenbankzugriff:** Die Applikation muss mit einer Schnittstelle zu einer Datenbank verbunden sein, da die Inspektionsdaten in einer Datenbank gespeichert werden.
- **Zugriff auf Hardware Schnittstellen:** Die App muss einen Zugriff auf die Kamera des mobilen Gerätes gewährleisten können.

Anhand dieser Kriterien war die Wahl der mobilen Entwicklungstechnologien zwischen Hybrid und webbasierten Entwicklungstechnologien. Grundlage beider Plattformen sind die Webtechnologien. Der Faktor, der ausschlagend war, ist die Anforderung, dass die App auch auf der Desktopoberfläche laufen soll. Deswegen wurde als mobile Entwicklungstechnologie, die **webbasierten Entwicklungstechnologien** ausgewählt. Noch einmal eine Zusammenfassung der ausschlaggebenden Gründe, für die Auswahl der webbasierten Entwicklungstechnologien:

- Durch den Aufruf über den Browser ist die Applikation plattformunabhängig.
- Es ist möglich, durch den Browser auf die Kamera des mobilen Gerätes zuzugreifen.
- In den Anforderungen wird kein natives Aussehen als Notwendigkeit definiert.
- Verbindung mit Datenbanksystem, durch den Einsatz von Webtechnologien kann problemlos auf Datenbanksysteme wie MySQL und MongoDB zugegriffen werden.
- In den Anforderungen war auch die Offline Funktionalität kein Kriterium.

Bei der Auswahl der Web Technologie war die eine der Voraussetzungen eine Verbindung mit einer Backend Technologie die einen performanteren Umgang mit der Datenbehandlung hat. Deswegen wurde die Wahl auf eine JS Frontend Bibliothek gesetzt, dadurch sind Backend und Frontend unabhängig voneinander. Nach den Ergebnissen der Performance Tests von Angular, React und Vue wurde das Frontend Framework Vue ausgewählt.

5.2.2 Systemarchitektur

Basierend auf Anforderungen und Entwicklungstechnologien wurde die Entwicklung der Systemarchitektur entwickelt. Die Architektur basiert auf dem Client-Server Prinzip. Die Benutzerin oder der Benutzer - also der Client der Applikation - ruft per Browser die URL der App auf. Über das HTTP Protokoll wird ein Request an die API Schnittstelle geschickt, diese greift dann auf die Ressource am Server und der organisiert dann die Datenverarbeitung. In der Abbildung 26 ist die Systemarchitektur der Applikation abgebildet.

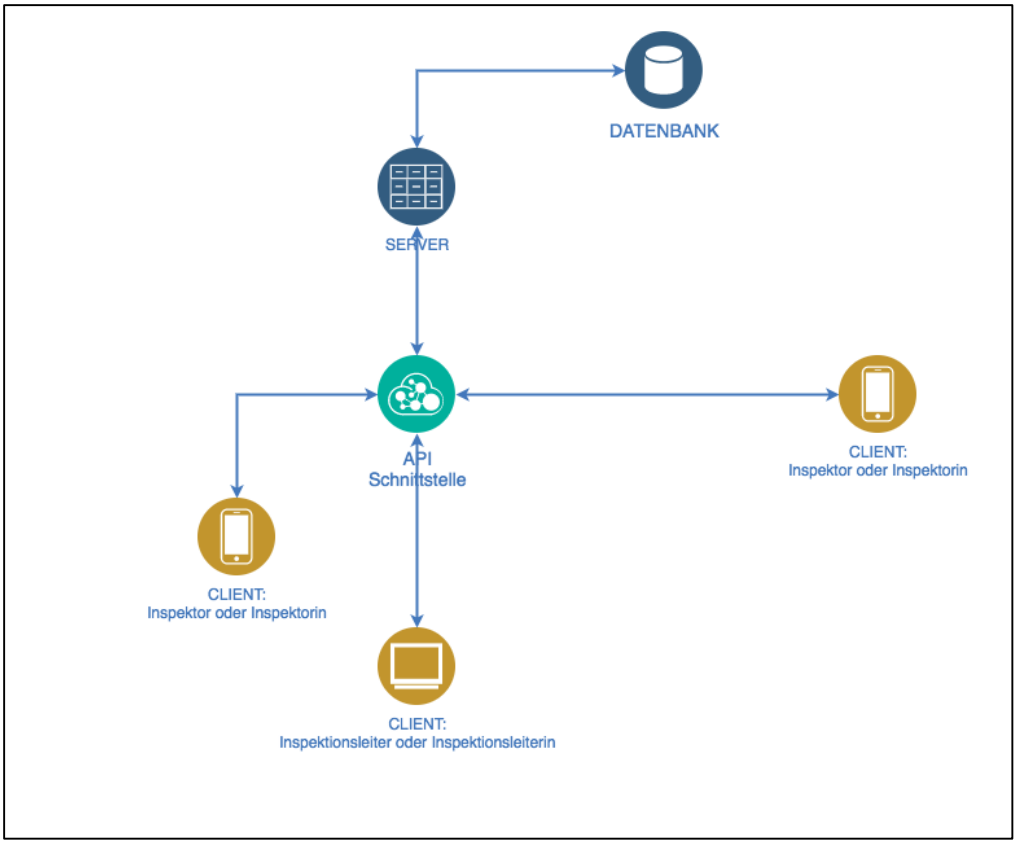


Abbildung 26: Systemarchitektur der Applikation

Vorrausetzung dieser Form der Architektur ist eine Internetverbindung. Die komplette Datenverarbeitung läuft über die API Schnittstelle, sowohl bei der Editierung, als auch bei der Beschaffung von Datensätzen.

5.2.3 Datenbanksystem

Bei der Auswahl des Datenbanksystems war die Auswahl zwischen einer rationalen und nicht-rationalen Datenbank. Der prominenteste Vertreter einer rationalen Datenbank im Bereich der Webtechnologien ist MySQL. Die Strukturierung von Daten erfolgt in Tabellen. Um größere Mengen von Daten in rationalen Datenbanken zusammen zu sammeln, werden meist die Daten über mehrere Tabellen referenziert (Parker, Poe, & Vrbsky, 2013).

Sie verringern auch Leistungsprobleme bei datenintensiven Anwendungen und können besser mit vielen Schreib- und Leseanfragen umgehen. Es gibt vier Kategorien von NoSQL-Datenbanken: Key-Value-Datenbanken, spaltenorientierte Datenbanken, Graph Datenbanken und dokumentenorientierte Datenbanken. (Bhogal & Choksi, 2015).

Im Bereich der nicht-rationalen Datenbanken hat MongoDB große Marktanteile. D Ein Merkmal ist, dass die Speicherung von Daten dokumentenbasiert erfolgt.

MongoDB wurde für eine hohe Skalierbarkeit und Verfügbarkeit entwickelt („MongoDB and MySQL Compared“, 2017). Vorteile gegenüber MySQL ist das dynamische Schema, bei MySQL ist die Datenstruktur vorgegeben. Bei MongoDB kann das das Datenschema ohne großen Aufwand erweitert werden. Die Vorteile von MySQL ist der performantere Umgang mit komplexeren Transaktionen.

In einem Performance Test zwischen MongoDB und MySQL wurden folgende Werte ermittelt und in der Tabelle 5 dargestellt. Der Test wurde auf einem Node.js Server durchgeführt (Caetano, 2015/2017).

Tabelle 5: Performance Test zwischen MySQL und MongoDB (Caetano, 2015/2017)

Parameter	***MySQL***	***MongoDB***
100 Einträge einfügen	*1702 ms*	*167 ms*
100 Einträge selektieren	*10 ms*	*60 ms*
1000 Einträge einfügen	*94813 ms*	*1013 ms*
1000 Einträge selektieren	*13 ms*	*677 ms*

Die Werte können verbessert werden, wenn die Datenbankkonfiguration noch optimiert werden würde. Aber die Testergebnisse lassen schon erkennen, dass der MySQL im Bereich der Selektierung von Daten performanter als MongoDB ist. Dafür hat MongoDB einen schnelleren Zugriff im Einfügen von Daten in die Tabelle.

Da der Faktor Performance in diesem Beispiel keine großen Aufschlüsse gab, wurde ein anderer Punkt als Hauptkriterium für die Auswahl des Datenbanksystems gewählt. Bei MongoDB ist die Skalierbarkeit besser handzuhaben und das Datenbankschema muss nicht vordefiniert werden. Dadurch bleibt für manche Anforderungen ein größerer Spielraum bei der Definition der Datenstruktur.

6 Prototyp

In diesem Kapitel wird der Aufbau und die Entscheidungskriterien des Prototyps beschrieben. Es wird auch erörtert, welche technischen Problemstellungen zu lösen waren und wie das Anforderungsprofil umgesetzt wurde.

6.1 Technik

Wie schon vorher erwähnt, ist die Wahl der mobilen Entwicklungsoptionen auf webbasierte Entwicklungstechnologien gefallen. Die Technik wird in zwei verschiedene Teile aufgeteilt: der Backend Teil und der Frontend Teil.

6.1.1 Backend Technologie

Der Backend Teil organisiert die Datenverarbeitung und stellt definiert die API-Schnittstelle. Außerdem wird durch das Backend, die Verbindung zur Datenbank hergestellt. Die Umsetzung wurde mit der Programmiersprache PHP durchgeführt, speziell wurde das Framework Laravel verwendet. PHP ist eine weit verbreitete und für den allgemeinen Gebrauch bestimmte Open Source-Skriptsprache, welche speziell für die Webprogrammierung geeignet ist und in HTML eingebettet werden kann (PHP Group, 2017). Laravel ist ein Framework, das nach der MVC Struktur agiert. Der große Nutzen von Laravel ist die Struktur, wie die Komponenten vordefiniert wurden, aber der Entwicklerin oder den Entwickler genug Handlungsfreiraum lässt.

Laravel bringt eine große Funktionspalette mit, um eine moderne Web Applikation umzusetzen. Durch den PHP Pakete Manager Composer wurden einige zusätzliche Erweiterungen installiert. Unter anderem ein Paket, das die Integration von MongoDB in das Framework ermöglicht.

6.1.2 Frontend Technologie

Als Frontend Technologie wurde das JavaScript Framework Vue integriert. Vue bekannt als Vue.js ist ein progressives JavaScript Framework, das dazu gedacht ist, um User Interfaces zu bauen. Vue ist ein sehr mächtiges Framework mit einer gut ausgebauten Funktionspalette, sehr leichtgewichtig in der Dateigröße und mit einer geringen Kurve zu erlernen.

Das komplette User Interface wurde mit Vue umgesetzt, die Kommunikation zwischen dem Frontend und Backend funktioniert mit der API Schnittstelle. Die

Datenverarbeitung findet im Backend Bereich statt. Das Frontend holt sich die Daten und stellt diese in den einzelnen Sektionen dar.

6.1.3 API Schnittstelle

Eine API Schnittstelle bietet auch eine gewisse Flexibilität. Das Interface könnte problemlos in einer Android oder iOS Applikation zur Datenverarbeitung integriert werden. Die Abbildung 27 veranschaulicht noch einmal in vereinfachter Darstellung, wie eine API Schnittstelle arbeitet.

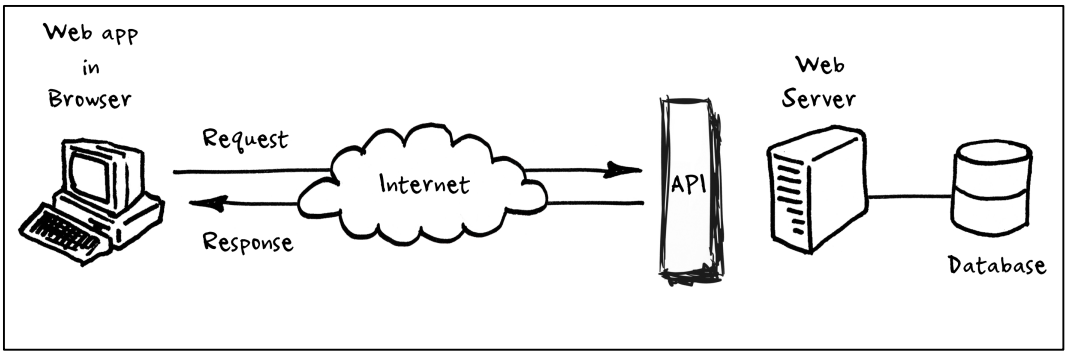


Abbildung 27: Funktionsabbildung einer API Schnittstelle

Wie in der Grafik zu erkennen ist, interagiert das User Interface mit POST und GET Aufrufen zur API Schnittstelle. Diese holt die Daten aus der Datenbank und sendet diese wieder per Response an das User Interface.

6.2 Umsetzung

6.2.1 Lokale Installation

Für eine ordnungsgemäße Entwicklungsumgebung wurde auf folgende Eigenschaften der Fokus gerichtet:

- Einrichten eines lokalen Web Servers - in diesem Fall wurde die Software XAMPP verwendet, die einen lokalen Apache Web Server mit der PHP Version 7.1 repräsentiert.
- Zusätzlich musste MongoDB installiert werden, diese Installation erfolgte via Ausführungsdatei von der Webseite des Herstellers.
- Um den Apache Web Server mit MongoDB zu verbinden, mussten einige Änderungen in der Konfigurationsdatei vorgenommen werden.
- Zusätzlich wurden die sogenannten Virtual Host Einträge zum Server hinzugefügt, damit die URL mit brdins.dev aufgerufen werden konnte. Der URL Name brdins war eine Kurzform für **brigdeinspection**.

- Weiters wurde das Tool Studio 3T am Rechner installiert. Studio 3T ist eine grafische Oberfläche, mit der die MongoDB verbunden wird(3T Software Labs GmbH, 2017). Dadurch ist es möglich, durch eine grafische Oberfläche die Dateneinträge der einzelnen Datenkollektionen anzusehen.

Diese Schritte waren erforderlich, um das PHP Framework Laravel am lokalen Web Server laufen zu lassen.

Jetzt wurde die Laravel Applikation erstellt, dazu wurde folgender Befehl in der Konsole ausgeführt:

Listing 1: Composer Befehl zur Erstellung eines neuen Laravel Projektes

```
composer create-project --prefer-dist laravel/laravel bridgeinspection
```

Jetzt wurde ein neues Laravel Projekt im Verzeichnis, wo dieser Befehl ausgeführt wurde, erstellt. Voraussetzung dafür ist, dass der PHP Paketmanager am Entwicklungssystem installiert ist.

Im Virtual Host Eintrage wurde der Pfad vom Laravel Verzeichnis eingetragen, dann konnte der Web Server gestartet und das Projekt per brdins.dev aufgerufen werden.

6.2.2 Setup von Git

In einer modernen Entwicklungsumgebung ist der Einsatz eines Versionsverwaltungssystems unumgänglich. Deswegen wurde zusätzlich ein Git Repository auf der Open Source Plattform GitHub eingerichtet.

Hinter Git steckt ein verteiltes Versionsverwaltungssystem. Es existiert zwar ein zentrales Repository, in welches sämtliche Änderungen einfließen, allerdings laden sich alle Benutzerinnen und Benutzer ihre eigene Arbeitskopie herunter. Dadurch haben sie das gesamte Repository inklusive der Historie lokal vorliegen und sind nicht auf eine durchgehende Verbindung zum Netzwerk angewiesen; zudem werden die Änderungen sehr schnell ins Haupt-Repository übertragen („Git vs. SVN – Versionsverwaltung im Vergleich“, 2016).

6.2.3 Implementierung von Vue

Laravel liefert bereits eine composer.json Datei mit. In dieser Datei werden die ganzen Pakete oder auch bezeichnet als *dependencies* des Node Package Managers aufgelistet.

Der Node Package Manager (npm) ist ein Paketmanager für die JavaScript-Laufzeitumgebung node.js (npm Inc., 2017). Im Laravel wird dieser Manger verwendet, um die Frontend Technologien zu verwalten. Alle Pakte die per npm installiert werden, laufen unter freier Lizenz.

Die benötige Vue Bibliothek wurde durch *npm* zum Projekt hinzugefügt, folgender Befehl wurde im Verzeichnis ausgeführt.

Listing 2: Vue Ressourcen wurden zum Projekt hinzugefügt

```
npm install vue
```

Zusätzlich wurden Pakete für das Routing, HTTP Aufrufe und für das Interface Design installiert.

Laravel bietet eine Erweiterung, die es möglich macht, den kompletten JavaScript Code in eine Datei zusammenzufassen und komprimiert diese auf eine minimale Dateigröße. Die Erweiterung heißt Elixir und arbeitet mit Gulp zusammen. Gulp oder gulp.js ist ein Webentwicklungstool, das basierend auf node.js, verschiedene Aufgaben automatisiert übernimmt. Dieses Tool wird auch als Build Tool beschrieben (Fedosejev & Bush, 2015).

Listing 3: Konfiguration des Build Systems

```
const elixir = require('laravel-elixir');

require('laravel-elixir-vue-2');


elixir(mix => {
    mix.sass('app.scss')
       .styles('../../../node_modules/vuetify/dist/vuetify.min.css')
       .webpack('app.js');
});
```

In diesem Code Teil werden die Styles Dateien und das JavaScript Material auf eine sehr geringe Dateigröße komprimiert. Über HTML Code werden diese Dateien dann eingebunden.

Nach diesen Konfigurationsschritten war die Entwicklungsumgebung mit ihren Komponenten aufgesetzt und die Applikation konnte entwickelt werden.

6.2.4 Konfiguration einer Single-Page-Webanwendung

Die Frontend Applikation wurde als Single-Page-Webanwendung programmiert. Der Unterschied zwischen klassischen, serverseitig gerenderten Webseiten und Single-Page-Webapplikation, liegt hauptsächlich in der geänderten Rolle des Servers. Während er ursprünglich neben dem Ausliefern der Ressourcen (HTML, CSS, JS) auch das komplette Rendering übernahm, ist er heute primär ein Datenlieferant (in Form von JSON) und stellt nur noch eine initiale Seite zur Verfügung (Viering, 2017).

Das heißt, die Inhalte der App werden dynamisch nachgeladen und haben den Vorteil, dass dadurch kaum Ladezeiten vorhanden sind. Der Grund dafür ist, weil die Applikation schon bei dem ersten Aufruf am Client gerendert wird.

Vue besitzt die Möglichkeiten diesen Typus einer Web Applikation umzusetzen. Das Routing wird mit Vue definiert und nicht über Laravel. Damit das automatisiert passiert, muss im Routing von Laravel folgendes Routing definiert sein.

Listing 4: Laravel Routing

```
<?php

// Instead of laravel routes, will use vue routes.
Route::get('{vue_capture?}', function() {
    return View::make('welcome');
})->where('vue_capture', '[\/\w\.-]*');
```

6.2.5 Dateistruktur der Applikation

Das MVC Laravel ist ein großes Framework, das viele Komponenten mitliefert. In der folgenden Auflistung werden die relevanten Ordner für den Prototyp beschrieben:

- **app:** Hier werden die Controller definiert - diese sind verantwortlich für die Logik der Applikation. Weiters werden auch hier die Models definiert. Die Struktur von den Daten, die später in die Datenbank hinzugefügt werden, wird nach der Definition des jeweiligen Models gehandhabt.
- **config:** Hier werden die Packages rund um die Authentifizierung und die Verbindung der Datenbank konfiguriert.
- **resources:** Hier wird das User Interface gebaut, die ganzen Vue Komponenten werden in diesem Ordner erstellt. Die kompletten Frontend Teile wie Routing und HTTP Aufrufe werden in diesem Ordnerteil verwaltet.
- **routes:** Hier wurde die Route für die Weiterleitung auf das Vue Routing definiert und zusätzlich werden die API Routen in einem eigenen File definiert.
- **.env:** Dieses File liegt im Hauptverzeichnis und hier werden alle wichtigen globalen Variablen definiert. Wie zum Beispiel die Datenbankverbindung inklusive Passwörter.

6.2.6 Datenbankstruktur

Als Datenbanksystem wurde in der App MongoDB verwendet. MongoDB ist eine NoSQL Datenbank, also eine nicht-rationale Datenbank. Die Datenablagerung findet in den sogenannten Kollektionen oder Collections statt. Im Laravel Framework wird durch die Models die Struktur der einzelnen Collections definiert.

Es werden Beziehungen zu anderen Collections und die Attribute der Collection im Model definiert. Die Datenabbildung passiert in folgenden Collections:

- **user:** Hier wird die Information zum Benutzer gespeichert. Zusätzlich wird definiert, ob der Account aktiviert ist oder nicht.
- **userVerfication:** Hier werden die Aktivierungscodes zu den einzelnen Benutzern gespeichert.
- **bridge:** In dieser Collection werden die Daten zu den Brücken erfasst.
- **inspection:** Hier werden alle Daten rund um die Brückeninspektion erfasst, in der Datenbankstruktur wurde die Beziehung zur Brücke mit 1:n definiert, d.h. eine Brücke kann mehrere Inspektionen haben.
- **image:** Hier werden zu den abgelegten Bildern am Server zusätzlich Einträge in der Datenbank verfasst. Beziehung zwischen inspection und image ist auch mit 1:n definiert, weil eine Inspektion mehrere Bilder haben kann.
- **imageMarker:** In dieser Collection werden die markierten Schäden am Bild erfasst. Die Beziehung zwischen image und imageMarker ist ach 1:n, weil ein Bild natürlich mehrere Schäden aufweisen kann.

In Abbildung 28 bekommt man einen Überblick wie die Datenbankstruktur aufgebaut ist.

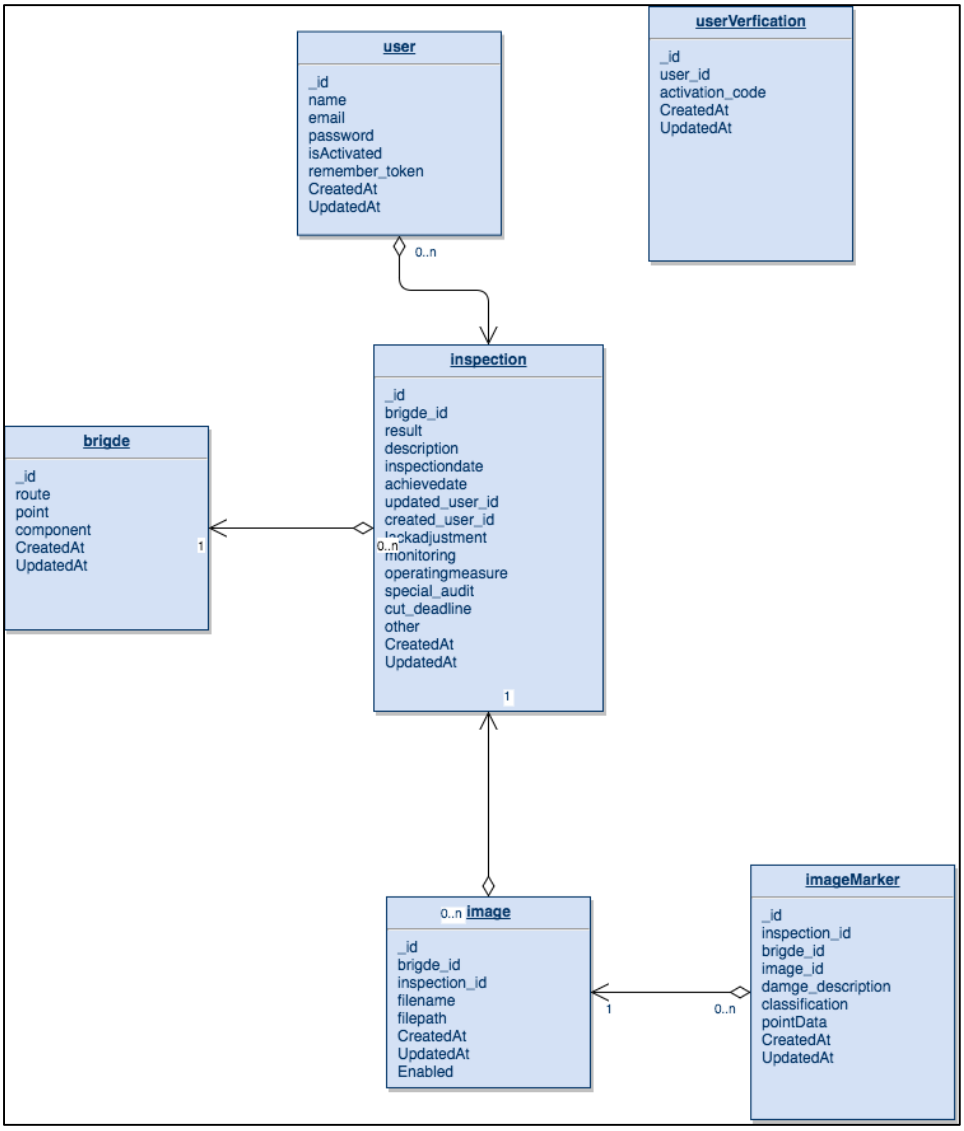


Abbildung 28: Datenbankstruktur der Applikation

6.2.7 Routing

6.2.7.1 Vue Routing

Was genau Routing ist, wurde im Punkt Begriffserklärung beschrieben. Da in der Webanwendung sehr viel mit eindeutigen Eigenschaften gearbeitet - mit sogenannten IDs – wird, müssen auch im Routing Platzhalter definiert werden.

Die Zuweisung zu speziellen Komponenten passiert per Aufruf der URL. Im Listing 5 wird eine Übersicht gegeben, wie der Seitenaufbau der Applikation aussieht:

Listing 5: Vue Routing bzw. URL Aufbau der Applikation

```
export default new VueRouter({
```

```
    mode: 'history',
    base: __dirname,
      routes: [
        { path: '/', name: 'home', component: bridgeOverview },
        { path: '/auth', name: 'authentication', component: auth },
        { path: '/bridges', name: 'bridges.show', component:
bridgeOverview },
        { path: '/bridge/add', name:'bridge.add', component: bridgeAdd
},
        { path: '/bridge/:id', name: 'bridge.show', props: true,
component: bridgeShow },
        { path: '/bridge/:id/edit', name: 'bridge.edit', props: true,
component: bridgeEdit },
        { path: '/bridge/:id/addinspection', name:
'bridge.addinspection', props: true, component: inspectionAdd },
        { path:
'/bridge/:id/inspection/:inspectionId/:fileId/addmarker', name:
'inspection.addmarker', props: true, component: imageMarker },
        { path: '/bridge/:id/inspection/:inspectionId/edit', name:
'inspection.edit', props: true, component: inspectionEdit }


      ]
});
```

Das Routing in der Applikation wird immer nach folgendem Schema definiert:

- **path:** Die URL, die die Userin oder der User aufruft, Platzhalter werden mit Doppelpunkt gesetzt.
- **name:** Der Name der URL. Dieser Parameter wird in Vue verwendet um auf andere Seiten weiterzuleiten.
- **component:** Hier wird auf das File verwiesen in dem der HTML Code projiziert wird.
- **props:** Ist der Parameter, wenn Eigenschaften an diese Route weitergegeben werden wie z.B.: IDs.

6.2.7.2 API Routing

Das API Routing wird in Laravel Routen definiert und in der App dann per HTTP Aufruf aufgerufen. Im Listing 6 werden ein paar Routen der API gezeigt und beschrieben:

Listing 6: API Routing

```
Route::prefix('api')->group(function () {
  Route::post('/bridge/add', 'BridgeController@store');
  Route::get('/bridge/all', 'BridgeController@showAll');
  Route::get('/bridge/{id}', 'BridgeController@show');
  Route::post('/bridge/{id}/delete', 'BridgeController@destroy');
  Route::get('/bridge/{id}/allInspections',
'InspectionController@showAllforBridge');
  Route::post('/bridge/{id}/edit', 'BridgeController@update');
  Route::post('/inspection/photos/upload', 'ImageController@store');
```

```
  Route::post('/inspection/add', 'InspectionController@store');
  Route::post('/inspection/{id}/edit', 'InspectionController@update');
});
```

Bei der Bestimmung der Routen ist es wichtig, den Typ des Requests anzugeben - POST, GET oder PUT. Die Platzhalter der Routen werden mit geschwungen Klammern definiert. Die zugewiesene Logik der Route wird per Controllername und mit dem @-Zeichen auf die benötigte Methode verwiesen.

6.2.8 Methodik der Templates

Das Rendern der HTML Templates übernimmt Vue, für jede Ressource der Applikation wurden Templates erstellt. Der Vorteil von Vue ist, dass es die Möglichkeit besitzt, die Templates ineinander zu verschachteln. Deswegen wurden z.B.: für den Image Upload eigene Templates definiert, die dann später in die Hauptkomponente eingebunden wurden. Im Listing 7 ist ein Beispiel aus der App wie die Handhabung mit den Subtemplates funktioniert:

Listing 7: Definition einer Textbox mit Fehlermeldung

```
<v-text-field box
  :label= "this.$t('bridge.name')"
  v-model= "newBridge.route"
>
</v-text-field>
<FormError v-if="errors.route" :errors="errors">
   {{ errors.route[0] }}
</FormError>
```

Die Daten der Textbox werden im Objekt newBridge in dem Attribut route gespeichert. Das Element FormError ist ein Subtemplate, das eingebunden wird, falls der Server eine Fehlermeldung zurückschickt, weil nicht valide Daten eingegeben wurden, wird im Element FormError ein Fehler angezeigt.

Im Listing 8 wird die gezeigt, wie die Fehlerbehandlung und das Absenden von Daten funktioniert.

Listing 8: Definition einer Textbox mit Fehlermeldung

```
createBridge() {
  axios.post('/api/bridge/add', this.newBridge)
   .then(response => {
     if(response.data.success) {
       this.newBridge = {
         route: '',
         point: '',
         component: ''
       }
```

```
        this.$popup({
            message         : this.$t('general.add'),
            color           : '#fff',
            backgroundColor : 'rgba(0, 0, 0, 0.7)',
            delay           : 5
        })
        this.dialog = true
        this.bridgeId = response.data.id
      }
      this.errors = response.data
    })
    .catch(e => {
      console.log(e)
    })
  },
```

Diese Funktionalität wird im Skriptteil im Template definiert, durch den HTTP Handler axios wird an die Schnittstelle das Objekt gesendet. Falls kein Fehler Response zurückkommt, erscheint ein Popup mit einer Erfolgsmeldung. Falls doch ein Fehler entstanden ist, werden diese in die Variable Errors gespeichert.

6.2.9 Logik der Datenbehandlung

Wie die Logik in der Datenbehandlung ist, wird im nächsten Listing gezeigt. Wichtig ist, dass Laravel zur Speicherung von Daten das PHP Erweiterung Eloquent verwendet. Das Paket ist eine Schnittstelle, die das PHP Backend mit der Datenbank zu verbindet.

Das Listing 9 zeigt, wie eine Brücke in die Datenbank gespeichert wird.

Listing 9: Speicherung einer Brücke in die Datenbank

```
public function store(Request $request)
{
  $validator = $this->validator($request->all());
  if ($validator->fails()) {
    return response()->json($validator->messages(), 200);
  }
  $bridgeItem = Bridge::create([
    'route' => $request['route'],
    'point' => $request['point'],
    'component' => $request['component']
  ]);
```

```
    return response()->json(['success' => 'success', 'id' =>
$bridgeItem->id], 200);
```

Die Validierungsfunktion überprüft, ob die gesendeten Daten vom Frontend nach dem Schema des Models definiert sind. In diesem Codeausschnitt sind alle drei Eigenschaften im Model als Pflichtfelder definiert. Wenn der Validierungsprozess fehlschlägt, wird ein Error zurückgegeben.

Im Erfolgsfall werden die Daten über das Model Bridge durch den Eloquent Builder in die Datenbank gespeichert. Zusätzlich wird eine Erfolgsmeldung zurückgeliefert.

6.3 Funktionalität

Großteils der Funktionalität wurde nach dem Schema der definierten Punkte im Thema Umsetzung abgebildet. In diesem Abschnitt werden die integrierten Funktionen von der Applikation beschrieben.

6.3.1 Brückenadministration

Der Benutzer oder die Benutzerin hat die Möglichkeit, Brücken anzulegen, zu editieren oder zu löschen. Die Eingabefelder im Anlegungsprozess oder in der Bearbeitungsansicht sind als Pflichtfelder deklariert, weil diese Daten unentbehrlich für die Inspektion sind.

Die Abbildung 29 zeigt den Hinzufügungsprozess einer Brücke, die Auflösung des Bildschirmes ist die Auflösung eines iPads im Landscape Modus.

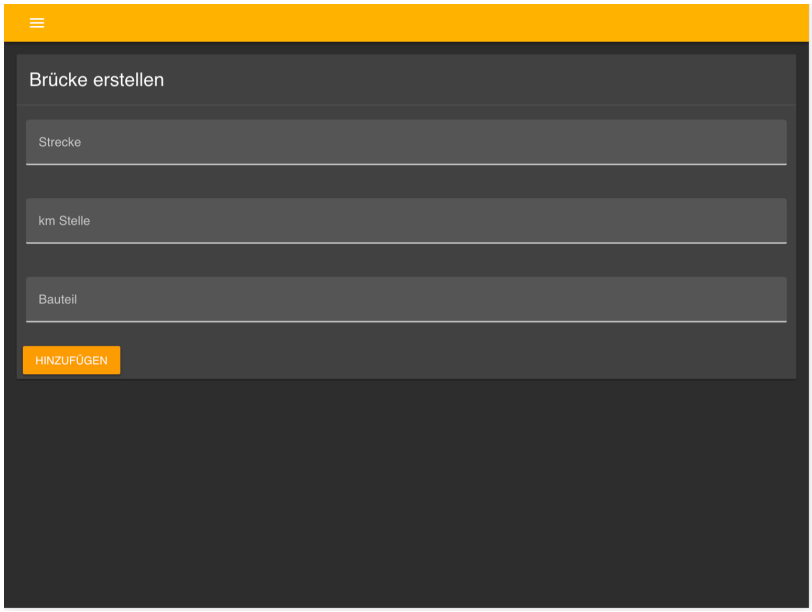


Abbildung 29: Abbildung von dem Screen "Brücke hinzufügen"

In der Abbildung 30 wird dargestellt, wie die Brückenübersicht aussieht. Es besteht die Möglichkeit, Brücken nach Namen, km-Stelle, Bauteil, Erstellungsdatum zu sortieren. Außerdem ist eine Suchmaske für die gezielte Suche einer Brücke implementiert.

Zusätzlich besteht die Möglichkeit, die Brücke mit ihren Inspektionsdaten sofort in dieser Ansicht zu exportieren.

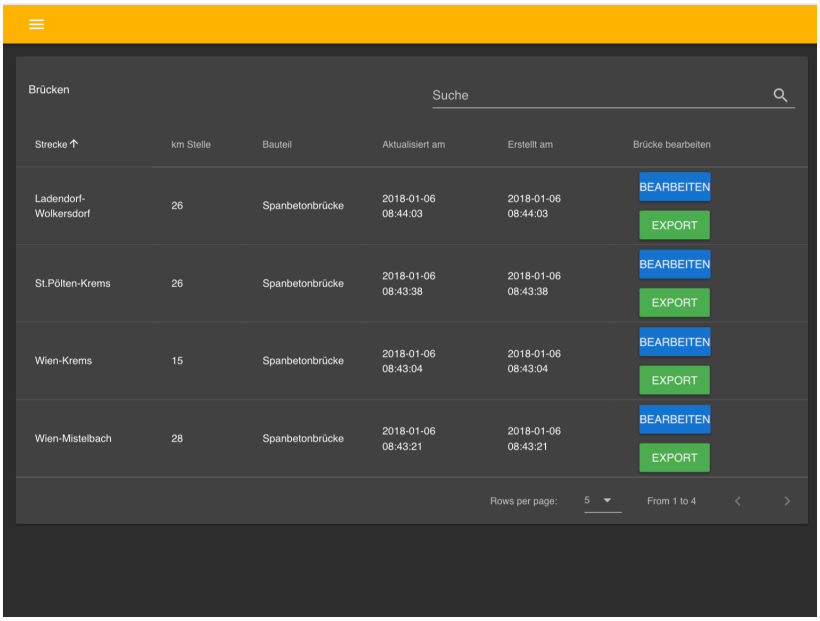


Abbildung 30: Brückenübersicht der Applikation

6.3.2 Inspektionsübersicht

Um in die Inspektionsübersicht zu navigieren, wählt die Benutzerin bzw. der Benutzer die Brücke aus. In der Abbildung 31 wird dargestellt, wie die Inspektionsübersicht am Tablet und auch am mobilen Endgerät, aussieht.

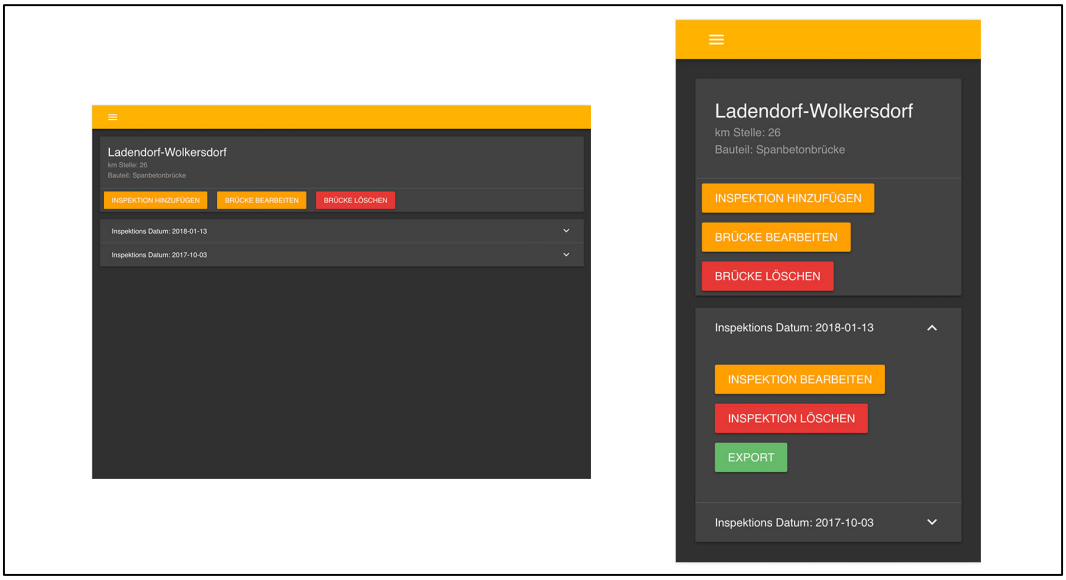


Abbildung 31: Inspektionsübersicht der Applikation

Durch einen Klick auf eine Inspektion kann die Userin oder der User das Menü aufklappen und die benötigte Funktionalität auswählen. Wie man in der Grafik erkennen kann, werden die Daten der Brücken noch einmal angegeben und zusätzlich die weitere Funktionspalette.

6.3.3 Hinzufügen von Inspektionen

Die Funktionalität im Prozess Hinzufügen von Inspektionen, wurde in drei Bereiche aufgeteilt.

6.3.3.1 Fotomaterial hinzufügen

Im ersten Bereich wurde die Funktionalität für das Hinzufügen von Fotomaterial implementiert. Die Benutzerin bzw. der Benutzer hat die Möglichkeit, mehrere Fotos zur Inspektion hochzuladen. Zusätzlich können Desktop Anwenderinnen oder Anwender per Drag and Drop, Bilder in die Applikation ziehen und diese werden dann hochgeladen. Mobile Anwenderinnen oder Anwender können auch über das Kamera Modul von ihrem Gerät direkt Fotomaterial hochladen.

In der Abbildung 32 wird der Prozess dargestellt, wie die Handhabung mit dem Hochladen des Fotomaterials funktioniert.

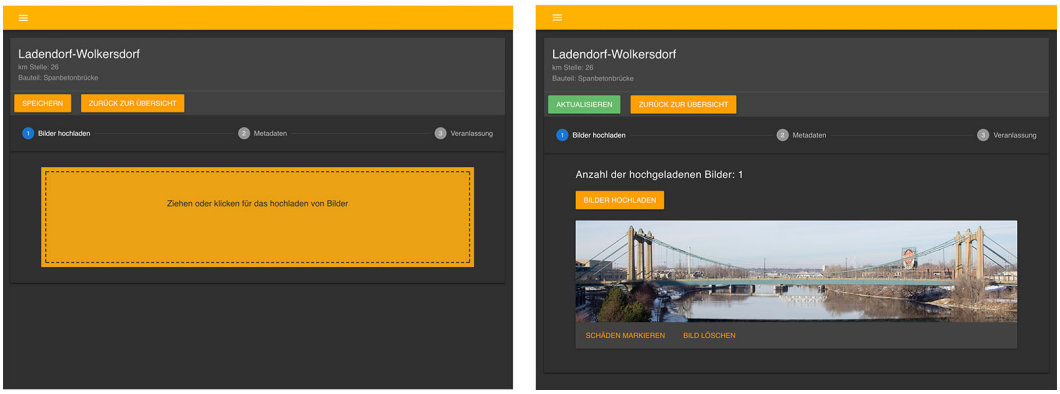


Abbildung 32: Darstellung der Funktionalität „Hochladen von Fotomaterial“

Hochgeladenes Fotomaterial, das möglicherweise für die Inspektion unbrauchbar ist, kann auch wieder gelöscht werden. Wenn die Anwenderin oder der Anwender Schadensmarkierungen am Bild vornehmen möchte, kann jetzt der Prozess dafür gestartet werden.

Es wurden auch Features integriert damit es keine Datenredundanzen gibt, wenn die Benutzerin oder der Benutzer vor dem Speichern Fotomaterial hochladet, wird automatisch eine Inspektion erstellt. Der Speichern Button wird dann automatisch durch den Aktualisierungsbutton ersetzt, damit die Anwenderin oder der Anwender die Information erhält, dass die Inspektion bereits angelegt ist.

6.3.3.2 Metadaten hinzufügen

Der zweite Teil des Inspektionsprozesses ist die Eingabe der Metadaten. Unter der Definition befinden sich folgende zwei Datumsangaben:

- **Inspektionsdatum:** Dieses Datum ist als Pflichtfeld deklariert. Beim automatisierten Prozess der Inspektionserstellung wird beim Hochladen von Fotomaterial, das heutige Datum generiert. Wenn die Anwenderin oder der Anwender keine Fotos hochlädt und probiert zu speichern, wird eine Fehlermeldung angezeigt, weil das Inspektionsdatum nicht ausgefüllt wurde.
- **Ausführungsdatum:** Dieses Datum wird gesetzt, falls eine Veranlassung getroffen wird.

6.3.3.3 Definition einer Veranlassung

Der dritte Bereich ist die Definition von Veranlassung. Die Anwenderin oder der Anwender kann wählen, welche Art von Veranlassung durchgeführt werden muss, um die Funktionalität der Brücke zu gewährleisten. In der Abbildung 33 ist zu erkennen, wie das User Interface mit dieser Anforderung umgesetzt wurde.

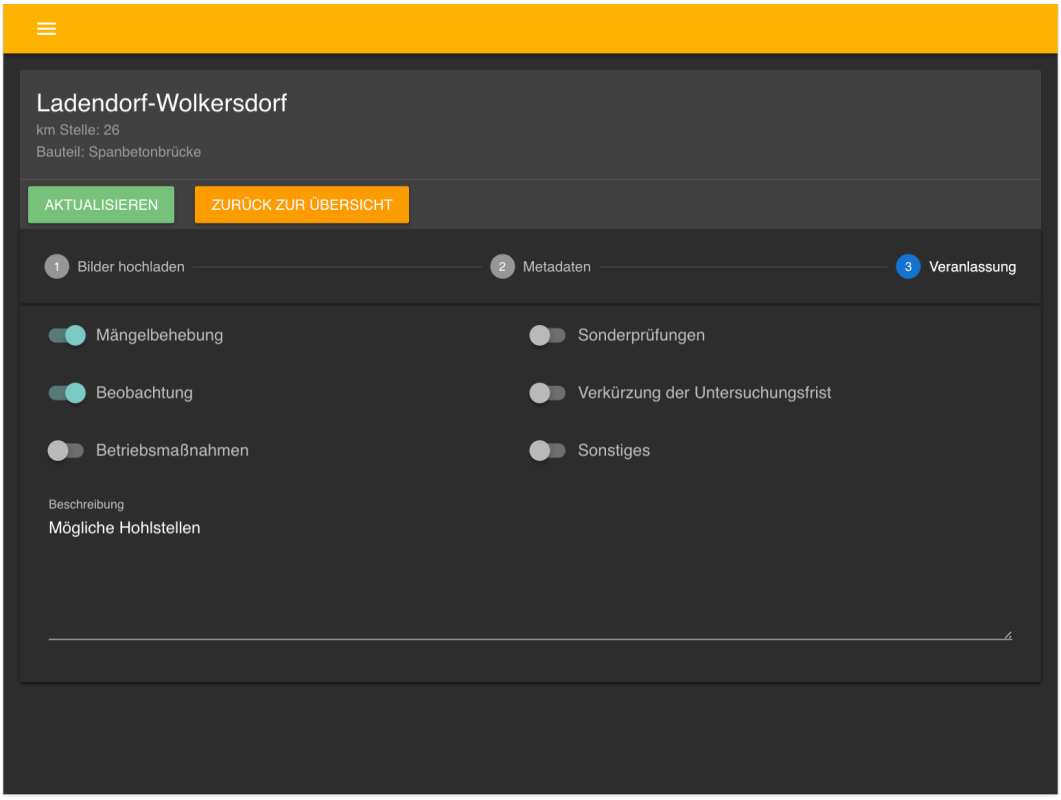


Abbildung 33: Setzen einer Veranlassung in der Applikation

Wenn ein Schieberegler gesetzt wird, erscheint automatisch das Textfeld in der die Beschreibung dazu verfasst werden kann.

6.3.4 Markieren von Schäden

Ein wichtiger Punkt in der Funktionalität ist das Markieren von Schäden. Die Schadensmarkierung kann nur bei Fotomaterial durchgeführt werden. Weiters werden Schäden wie im Anforderungsprofil vorgegeben, mit zwei Eigenschaften definiert. Schadensbeschreibung und Schadensklassifizierung, beide Felder sind als Pflichtfeld definiert und müssen von der Anwenderin oder vom Anwender ausgefüllt werden.

Es gibt zwei Möglichkeiten Schäden zu markieren, entweder durch den Button oder durch Klick auf die beschädigte Stelle im Bild. Das heißt, es wird keine Referenz zu einer Koordinate im Bild benötigt.

In der Abbildung 34 ist ein Beispiel mit zwei markierten Testschäden. Die roten Punkte sind die markierten Stellen.

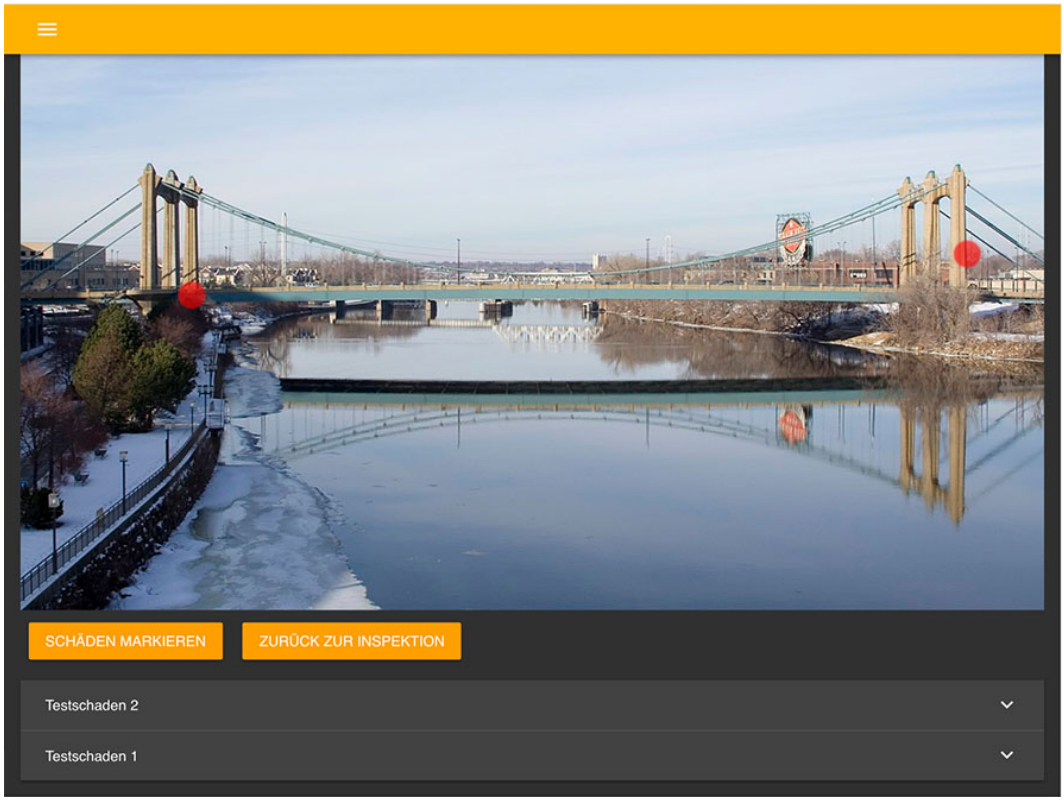


Abbildung 34: Screen der Schadensmarkierung in der Applikation

Durch einen Klick auf den Schaden im Menü oder am Bild, können die Daten noch einmal bearbeitet werden. Nach Bedarf kann der Schaden auch gelöscht werden. Wenn Schäden eingetragen wurden, werden diese in der Inspektionsübersicht dargestellt.

6.4 Abgleich Anforderungsprofil und Funktionspalette

Dieses Kapitel behandelt die Anforderungen, die nicht in die Entwicklung des Prototyps integriert wurden. Folgende Funktionalitäten wurden nicht im Prototyp umgesetzt:

- **Benutzerverwaltung:** Die Datenbankstruktur wurde definiert und die Funktionalität wurde auch ausprogrammiert. Dadurch, dass keine genaue Beschreibung im Bereich der Registrierung definiert wurde, war diese Funktionalität für den User Test nicht relevant.
- **Archivierungsmöglichkeit:** Die Anwenderin oder der Anwender des User Tests hat die Möglichkeit, den Export Button zu betätigen, aber die Funktionalität dahinter ist nicht ausprogrammiert.

- **Markieren von Schäden:** Diese Funktionalität wurde ausprogrammiert, aber es waren noch eine große Anzahl von Bugs und Fehlern implementiert. Deswegen wurde nur die einfache Variante über Textfelder in den Prototyp integriert.

7 User Tests

7.1 Ablauf und Konzept

Um den entwickelten Prototypen auf Usability und Funktionalität zu prüfen, wurde ein Konzept für User Tests entwickelt. Die Zielgruppe der Applikationen sind Personen, die im Bereich der Brückeninspektion arbeiten. Deswegen wurde bei der Ausarbeitung des User Tests der Fokus auf diese Zielgruppe gelegt. Es wurde ein Leitfadeninterview mit drei Testpersonen durchgeführt. Die Expertinnen und Experten mussten aufgabenbasierte Tests durchführen.

Zur Analyse der User Tests wurden ein Screenrecording und ein Fragenbogen verwendet. Der User Test wurde in drei Themen aufgeteilt:

- **Allgemeine Fragen:** Fragen zur Person, Alter, Geschlecht, Ausbildung und weitere demographische Fragen.
- **Testaufgaben:** In jeder der Testaufgabe wurde ein Teil der Funktionalität des Prototyps getestet.
- **Spezifische Fragen:** Zum Abschluss wurden der Testperson noch spezifische Fragen zur Applikation gestellt. Die Antwortmöglichkeit war mit Stichworten oder in ganzen Sätzen.

Nach dem die Testperson eine Aufgabe abgearbeitet hatte, mussten im Fragebogen folgende Fragen zur Aufgabe beantwortet werden:

- **Praxisbezug:** Die Testperson musste beantworten, ob diese Aufgabenstellung einen Praxisbezug hatte oder nicht.
- **Handhabung:** Die Testanwenderin oder der Testanwender musste auf einer Skala von 1-5 bewerten, wie umgänglich die Funktionalität in der App abgebildet wurde.

Die zehn Aufgabenstellungen deckten die komplette Funktionspalette der Applikation ab. Das Leitfadeninterview wurde als Anhang der Diplomarbeit bereitgestellt.

Die Ergebnisse der Expertinnen und Experten der Leitfadeninterviews werden alle einzeln präsentiert, weil jeder der Testpersonen verschiedene Interpretationen geliefert hat. In einer tabellarischen Ansicht werden die Handhabung bzw. die Usability der Funktion beschrieben. Außerdem werden die Verbesserungsvorschläge und Anmerkungen in den Kapiteln Interpretation der Ergebnisse aus den spezifischen Fragen und Interpretation der Ergebnisse der aufgabenbasierten Tests zusammengefasst.

7.2 Durchführung und Ergebnisse

Die Expertinnen und Experten im Bereich der Brückeninspektion bekommen die Bezeichnung als **TP1**, **TP2** und **TP3** in weiterer Folge. In der Tabelle 6 werden die demographischen Daten zu den Testpersonen angegeben. Die User Tests haben auf einem iPad 2016 stattgefunden und wurden an folgenden Tagen durchgeführt:

- **TP1:** 08.01.2018 in einer Büroumgebung
- **TP2:** 20.12.2017 in einer Büroumgebung
- **TP3:** 20.12.2017 in einer Büroumgebung

Tabelle 6: Demographische Daten der Testpersonen

	TP1	***TP2***	***TP3***
Alter	*52*	*44*	*43*
Geschlecht	*Männlich*	*Weiblich*	*Männlich*
Beruf	*Ziviltechniker Bauwesen*	*Bauingenieurin*	*Bauexperte*
Ausbildung	*Universitätsabschluss*	*Universitätsabschluss, Doktortrat Studium*	*Universitätsabschluss*
Mediennutzung Smartphone/Tablet	*Samsung(Android)*	*HTC(Android)* *iPad(iOS)*	*HTC(Android)*
Kenntnisse über den Prozess der Brückeninspektion	*Ja*	*Ja*	*Ja*
Erfahrung im Bereich der Brückeninspektion	*Im Rahmen der normalen Berufsausübung*	*Ingenieurbüro* *TU Wien*	*ASFINAG* *BMVIT*

Alle Expertinnen und Experten haben bereits beruflich im Bereich der Brückeninspektion gearbeitet bzw. arbeiten noch in diesem Bereich.

In der Tabelle 7 werden die Ergebnisse der aufgabenbasierten Tests dargestellt. Die Bewertungsskala ist von 1 bis 5: 1 bedeutet Sehr Gut und 5 bedeutet Nicht Genügend. Wenn in der Bewertung die Zahl 6 eingetragen wurde, hatte für die

Expertin oder den Experten die Aufgabe keinen Praxisbezug und deswegen wird die Funktionalität für diese Aufgabe nicht bewertet.

Tabelle 7: Ergebnisse der aufgabenbasierten Tests

	TP1	***TP2***	***TP3***
Aufgabe 1	*1*	*2*	*2*
Aufgabe 2	*1*	*1*	*1*
Aufgabe 3	*3*	*4*	*1*
Aufgabe 4	*3*	*4*	*2*
Aufgabe 5	*1*	*4*	*2*
Aufgabe 6	***6***	*4*	*2*
Aufgabe 7	***6***	*1*	*1*
Aufgabe 8	***6***	*2*	*2*
Aufgabe 9	*2*	*1*	*1*
Aufgabe 10	*1*	*2*	*2*

7.2.1 Interpretation der Ergebnisse der aufgabenbasierten Tests

Wie man in Tabelle 7 erkennten kann, ist eine hohe Schwankungsbreite bei manchen Testaufgaben. Der Fokus der Interpretation wird auf die Aufgaben gelegt, die bei den User Tests weniger gut abgeschnitten haben. Dazu wurden die Interpretationen separiert und die Ergebnisse der drei Testpersonen einzeln ausgewertet.

7.2.1.1 Testperson 1

TP1 hat die meiste Erfahrung im Bereich der Brückeninspektion, weil diese Testperson in diesem Berufsfeld seit Jahren tätig ist. TP1 hat folgende Aufgaben als nicht praxisrelevant beurteilt:

- **Funktionalität - Löschen von älteren Inspektionen:** Die Interpretation dieses Ergebnisses ist, dass es nicht relevant ist, ältere Daten von Brücken zu löschen, weil die Anwenderin oder der Anwender eine komplette

Chronologie von Inspektionsdaten haben möchte. Deswegen hatte diese Funktionalität für TP1 keinen Praxisbezug.

- **Funktionalität - Löschen von Markierungen:** Auch hier trifft die Interpretation über fehlende Daten in der Chronologie zu. Anmerkung der TP1 war: falls Fehler bei der Brückeninspektion entstanden sind, werden diese in der Nachbearbeitung und Beurteilung des Bauobjekts ausgebessert.
- **Funktionalität - Nachtrag von Inspektionsdaten:** Die Interpretation zu diesem Ergebnis ist, wenn eine ordnungsgemäße Historie zu Inspektionsmaterial besteht, müssen keine älteren Daten nachgetragen werden. Anmerkung von TP1 war, das älteres Inspektionsmaterial nur als Verweis angegeben werden soll und nicht in das System nachgetragen.

Alle anderen Aufgaben waren für TP1 praxisrelevant und die Handhabung der Funktionalität wurde mit positiven Noten beurteilt.

7.2.1.2 Testperson 2

TP2 hat Erfahrung im Bereich der Brückeninspektion und kennt den Prozess einer Inspektion. Der Interpretation zu Folge konnte TP2, mit den Aufgabenstellungen, die sich mit der Funktionalität „Markieren von Schäden“ beschäftigten, Problematiken im Bereich der Usability erkennen. Für die TP2 war es verwirrend, wie die Systematik zwischen Inspektion und Schadensmarkierung aufgebaut wurde. TP2 wünschte sich ein intuitiveres Design zu diesen Funktionalitäten.

Eine weitere Anmerkung zu den Aufgabenstellungen war die Definition von Bezeichnungen und Begriffen in der Applikation. Zusätzlicher Einwand war auch, dass bei Veranlassungen das Ausführungsdatum zur Veranlassung sichtlicher definiert werden sollte.

7.2.1.3 Testperson 3

TP3 hat Erfahrung im Bereich der Brückeninspektion und ist mit dem Prozess der Inspektion auch vertraut. TP3 war mit der Funktionsplatte und Usability zufrieden. Deswegen wurden überwiegend gute Noten für die Handhabung der Applikation vergeben.

7.2.2 Interpretation der Ergebnisse aus den spezifischen Fragen

Die Interpretation der Ergebnisse der spezifischen Fragen wurde einzeln zu den Testpersonen aufgearbeitet. Einige Faktoren können in der Interpretation definiert werden wie:

- **Tauglichkeit für praktische Brückeninspektion:** Alle Testpersonen haben eine positive Antwort abgegeben.

- **Verbesserungsvorschläge:** Jede Testperson hat mögliche Verbesserungsvorschläge für die Applikation definiert. Das heißt, der Prototyp hat noch einige Möglichkeiten für Verbesserung in diversen Bereichen und ist nicht in der Funktionalität limitiert.

7.2.2.1 Testperson 1

TP1 hatte eine Anmerkung im Bereich der Funktionalität zur Verwaltung von Fotomaterial. Die Grundidee war, dass es ein Übersichtsbild gibt und das in diesem Übersichtsbild einzelne Detailbilder verlinkt werden. Zusätzlich soll es dazu die Möglichkeit zur Schadensbeschreibung geben und die Kategorisierung zum Detailbild möglich sein.

Eine weitere Anmerkung war die Verwaltung von Planmaterial: die Anwenderin oder der Anwender soll die Möglichkeit haben, Planungsmaterial zur Brücke hochzuladen zu können.

Weiters hatte TP1 die Wichtigkeit der Datenstruktur und die Langlebigkeit der Applikation inklusive Datenbank angemerkt.

7.2.2.2 Testperson 2

TP2 hatte einige Anmerkungen im Bezug zur Usability in einigen Funktionalitäten, besonders im Bereich der Aufnahme von Schäden. Außerdem waren Anmerkungen für Änderungen der Bezeichnungen wie z.B.: bei der Brückenbeschreibung sollte Bauteil in Baustoff geändert werden.

Ein zusätzlicher Verbesserungsvorschlag war die Untersuchung des Prozesses der Straßenbrückeninspektion. Ziel dieser Anmerkung war es, unterschiedliche Prozesse kennenzulernen und ein allgemeines Pattern für den Inspektionsprozess zu definieren.

7.2.2.3 Testperson 3

TP3 hatte Verbesserungsvorschläge im Bereich der Aufnahme von Schadensmarkierungen. Die Testperson hatte für das Definieren von Schadensmustern folgende Idee: Die Anwenderin oder der Anwender der Applikation soll die Möglichkeit haben, einen Schaden direkt aus dem Schadenskatalog auswählen zu können. Außerdem wäre eine zusätzlich gewünschte Funktionalität, die Verknüpfung mit Planmaterial.

Eine weitere Anmerkung war die Berücksichtigung der Bedingungen vor Ort. Mögliche unangenehme Naturbedingungen wie Regen, Wind und Schnee erschweren die Inspektion und sollten im Anforderungsprofil definiert sein.

7.2.3 Mögliche Limitationen

Eine Erkenntnis aus den Ergebnissen der Leitfadeninterviews war, dass bezüglich von Verbesserungsvorschlägen, sehr unterschiedliche Ergebnisse geliefert worden sind. Das bedeutet, dass das Potential für Verbesserungen und Weiterentwicklungen gegeben ist und die Prozessverbesserung in dieser Form noch entwicklungsfähig ist.

Laut einer Studie, findet eine einzelne Expertin oder Experte nur 35 Prozent möglicher Problemstellungen. Aber bei einer Anzahl von drei bis fünf Expertinnen oder Experten werden 75 Prozent der Schwachstellen gefunden (Nielsen, 1992).

Wenn diese Studie die Richtlinie für die Leitfadeninterviews in Betrachtung zieht, ist zu erkennen, dass eine gewisse Limitation bei der Suche von Schwachstellen und Fehlerquellen der Applikation entstanden ist.

8 Fazit

In dieser Arbeit wurde untersucht, wie der Einsatz von mobilen Entwicklungstechnologien den Brückeninspektionsprozess unterstützen oder optimieren kann.

Die folgenden fünf Forschungsfragen wurden zu diesem Thema ausgearbeitet.

F1: Wie kann eine Prozessoptimierung bei einer Brückeninspektion aussehen und wie können mobile Anwendungen in der Entwicklung in diesem Themengebiet eingesetzt werden?

Im Kapitel Theoretische Ausgangslage wurden Analysen im Bereich von Forschungsprojekten und Konkurrenzanalysen durchgeführt. In den Ergebnissen waren Ansätze und Ideen, wie die Möglichkeit gegeben ist, den Inspektionsprozess mit Entwicklungstechnologien zu verbessern. Die Thematik mobile Entwicklungsmethoden und Brückeninspektionen war ein unerforschtes Gebiet.

Deswegen ist die Frage durch das Kapitel Theoretische Ausgangslage nicht zu beantworten gewesen. Deswegen musste eine Problemanalyse, Definition eines Anforderungsprofils und die Entwicklung eines Prototyps stattfinden um diese Forschungsfrage zu beantworten. Eine Prozessoptimierung sah folgendermaßen aus: durch die Unterstützung von mobilen Entwicklungstechnologien kann die Brückeninspektion am Smartphone oder Tablet stattfinden. Sämtliche Funktionalitäten und Abbildungen einer Prozessoptimierung sind in dem Kapitel Prototyp definiert.

F2: Wie sieht ein konkreter Prozess für eine Brückeninspektion aus?

Die Definition des Brückeninspektionsprozesses wurde im Kapitel Problemanalyse erforscht. Grundsätzlich bestand die Methodik für die Erkenntnis des konkreten Prozesses aus zwei Teilen: eines Expertinnen Interviews und einer Dokumentenanalyse. Bei einfacher Darstellung sieht ein Brückeninspektionsprozess folgendermaßen aus: Vorbereitung, Durchführung, Dokumentation und Nachbereitung.

F3: Wie empfinden Experteninnen und Experten die Handhabung und den Umgang mit der Applikation?

Um den Prototypen auf Funktionalität und Handhabung zu testen, wurden mit drei Expertinnen und Experten Leitfadeninterviews durchgeführt. Diese Interviews beinhalten aufgabenbasierte User Test, dadurch wurden mögliche Schwachstellen

und Probleme des Prototyps ermittelt. Die Schlussfolgerung ist, dass die Funktionalität inklusive Handhabung in großen Teilen positiv bewertet wurde.

Die Evaluierung der User Tests wurde im Kapitel User Tests präsentiert und erörtert. Hier sind auch die Anmerkungen für Optimierungen und Verbesserungen der Testpersonen definiert.

F4: Welche Anforderungsoptionen und Entwicklungsoptionen müssen für eine solche Applikation für die Umsetzung gegeben sein?

Die Ergebnisse im Kapitel Problemanalyse waren die Grundlage für die Erhebung der Anforderungsoptionen im Kapitel Anforderungen und technische Entscheidungen. Zusätzlich wurde, um einen Überblick über mobile Technologien zu bekommen, im Kapitel Mobile Entwicklungstechnologien eine State of the Art Analyse gemacht. Diese Inhalte sind dann in den Entscheidungsprozess bei der Wahl der Entwicklungstechnologie geflossen.

Ziel bei der Definition der Anforderungsoptionen war, dass die definierten Optionen in einer Applikation durch technische Mittel umgesetzt werden können. Wichtig bei der Auswahl der Entwicklungsoptionen war es, die Wahl einer Technologie mit der die Möglichkeit gegeben war, die komplette Palette an Anforderungen abzudecken. Die Wahl fiel auch eine JS Frontend Bibliothek mit einem PHP Backend.

F5: Welche Vorrausetzungen müssen erfüllt werden, um dem Anforderungsprofil und der Prozessanalyse gerecht zu werden?

Wichtig bei dieser Forschungsfrage ist, welche Mittel verwendet wurden, um eine qualitative Prozessanalyse zu machen. Das Anforderungsprofil ist aufbauend auf der Prozessanalyse. Deswegen war es wichtig, Kriterien zu finden, um eine qualitativ hochwertige Prozessanalyse zu machen. Vorrausetzungen waren, mehrere Kriterien einfließen zu lassen, deswegen wurde ein Expertinnen Interview und eine Dokumentenanalyse gemacht.

Eine weitere Vorrausetzung war, den Prototypen der Applikation zu testen und zu evaluieren. Damit konnte festgestellt werden, welche Schwerpunkte des Anforderungsprofils sinnvoll in die Applikation integriert wurden. Die User Tests brachten auch die Erkenntnis, welche Funktionalitäten nicht einwandfrei funktionierten und wo noch mögliche Schwachstellen sind.

Literaturverzeichnis

3T Software Labs GmbH. (2017). Studio 3T - The GUI for MongoDB. Abgerufen 4. Januar 2018, von https://studio3t.com/

Adobe. (2017a). Adobe AIR | Applikationen plattform- und geräteübergreifend bereitstellen. Abgerufen 3. Dezember 2017, von http://www.adobe.com/de/products/air.html

Adobe. (2017b). PhoneGap. Abgerufen 2. Dezember 2017, von https://phonegap.com/

Adobe. (2017c). Was ist Adobe PDF? PDF, Adobe PDF | Adobe Acrobat DC. Abgerufen 3. Januar 2018, von https://acrobat.adobe.com/at/de/why-adobe/about-adobe-pdf.html

Ahti, V., Hyrynsalmi, S., & Nevalainen, O. (2016). An Evaluation Framework for Cross-Platform Mobile App Development Tools: A case analysis of Adobe PhoneGap framework (S. 41–48). Gehalten auf der Proceedings of the 17th International Conference on Computer Systems and Technologies 2016, ACM. https://doi.org/10.1145/2983468.2983484

Ali, M., & Mesbah, A. (2016). Mining and characterizing hybrid apps (S. 50–56). Gehalten auf der Proceedings of the International Workshop on App Market Analytics, ACM. https://doi.org/10.1145/2993259.2993263

Apache. (2017). Apache CouchDB. Abgerufen 6. Dezember 2017, von http://couchdb.apache.org/

Appcelerator. (2012, November 21). Mobile App Development & MBaaS Products. Abgerufen 2. Dezember 2017, von http://www.appcelerator.com/mobile-app-development-products/

Apple. (2017a). App Store Review Guidelines - Apple Developer. Abgerufen 30. November 2017, von https://developer.apple.com/app-store/review/guidelines/

Apple. (2017b). Swift - Apple Developer. Abgerufen 30. November 2017, von https://developer.apple.com/swift/

Apple. (2018). Navigation Bars - Bars - iOS Human Interface Guidelines. Abgerufen 11. Januar 2018, von https://developer.apple.com/ios/human-interface-guidelines/bars/navigation-bars/

Bhogal, J., & Choksi, I. (2015). Handling Big Data Using NoSQL. In *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops* (S. 393–398). https://doi.org/10.1109/WAINA.2015.19

Bundesministerium für Verkehr, Bau und Stadtentwicklung. (2013). Bauwerksprüfung

nach DIN 1076 Bedeutung, Organisation, Kosten.

Caetano, A. (2017). *mongo-mysql: Mongo vs Mysql Test Performance in Nodejs*. JavaScript. Abgerufen von https://github.com/webcaetano/mongo-mysql (Original work published 2015)

Corral, L., Sillitti, A., & Succi, G. (2012). Mobile Multiplatform Development: An Experiment for Performance Analysis. *Procedia Computer Science*, *10*, 736–743. https://doi.org/10.1016/j.procs.2012.06.094

Czichos, H., & Habig, K.-H. (2010). *Tribologie-Handbuch Tribometrie, Tribomaterialien, Tribotechnik*. Wiesbaden: Vieweg+Teubner Verlag / GWV Fachverlage, Wiesbaden. Abgerufen von http://dx.doi.org/10.1007/978-3-8348-9660-5

Definition » Customer Relationship Management (CRM) « | Gabler Wirtschaftslexikon. (o. J.). Abgerufen 29. November 2017, von http://wirtschaftslexikon.gabler.de/Definition/customer-relationship-management-crm.html

Deveria, A. (2017). Can I use... Support tables for HTML5, CSS3, etc. Abgerufen 13. Dezember 2017, von https://caniuse.com/#search=service%20workers

Easyrec. (2017, April 23). In *Wikipedia*. Abgerufen von https://en.wikipedia.org/w/index.php?title=Easyrec&oldid=776779256

Facebook. (2017). React Native | A framework for building native apps using React. Abgerufen 30. November 2017, von https://facebook.github.io/react-native/index.html

Fedosejev, A., & Bush, A. (2015). *React.js essentials: a fast-paced guide to designing and building scalable and maintainable web apps with React.js*. Abgerufen von http://public.eblib.com/choice/publicfullrecord.aspx?p=3564808

FH St.Pölten. (2017). Dipl.-Ing. Dipl.-Ing. Dr. Hirut Grossberger — Fachhochschule St. Pölten. Abgerufen 14. Dezember 2017, von https://www.fhstp.ac.at/de/uber-uns/mitarbeiter-innen-a-z/grossberger-hirut

Funken, C., & Schulz-Schaeffer, I. (Hrsg.). (2008). *Digitalisierung der Arbeitswelt: zur Neuordnung formaler und informeller Prozesse in Unternehmen* (1. Aufl). Wiesbaden: VS Verlag für Sozialwissenschaften.

Gaunt, M. (2017). Service Workers: an Introduction | Web Fundamentals. Abgerufen 13. Dezember 2017, von https://developers.google.com/web/fundamentals/primers/service-workers/

Geetha, G., Safa, M., Saranya, G., & Subburaj, R. (2017). An effective practices, strategies and technologies in the service industry to increase customer loyalty using map indicator (S. 1–6). IEEE. https://doi.org/10.1109/ICIOTA.2017.8073607

Git vs. SVN – Versionsverwaltung im Vergleich. (2016, Juni 29). Abgerufen 4. Januar 2018, von https://www.1und1.at/digitalguide/websites/web-entwicklung/git-vs-svn-versionsverwaltung-im-vergleich/

Google. (2017a). Android. Abgerufen 30. November 2017, von https://www.android.com/

Google. (2017b). Checkliste für die Einführung | Android Developers. Abgerufen 30. November 2017, von https://developer.android.com/distribute/best-practices/launch/launch-checklist.html

Google. (2017c). Google Trends. Abgerufen 7. Dezember 2017, von /trends/explore

Google. (2018). Navigation with Back and Up | Android Developers. Abgerufen 11. Januar 2018, von https://developer.android.com/design/patterns/navigation.html

Griffith, C. (2017). *Mobile App Development with Ionic 2: Cross-Platform Apps with Ionic, Angular, and Cordova*. O'Reilly Media, Inc.

Ionic. (2017). Ionic Framework - About. Abgerufen 6. Dezember 2017, von https://ionicframework.com/about

Korotya, E. (2017, Januar 19). 5 Best JavaScript Frameworks in 2017. Abgerufen 10. Januar 2018, von https://hackernoon.com/5-best-javascript-frameworks-in-2017-7a63b3870282

Krause, S. (2018). *js-framework-benchmark: A comparison of the perfomance of a few popular javascript frameworks*. JavaScript. Abgerufen von https://github.com/krausest/js-framework-benchmark (Original work published 2015)

Lawson, N. (2015). Introducing Pokedex.org: a progressive webapp for Pokémon fans — Pocket JavaScript. Abgerufen 10. Januar 2018, von http://www.pocketjavascript.com/blog/2015/11/23/introducing-pokedex-org

Liebel, C. (2017, Juni 5). Progressive Web Apps, Teil 1: Das Web wird nativ(er). Abgerufen 13. Dezember 2017, von https://www.heise.de/developer/artikel/Progressive-Web-Apps-Teil-1-Das-Web-wird-nativ-er-3733624.html

Lynch, M. (2017, Januar 9). Ionic's 2016 - By the Numbers. Abgerufen 6. Dezember 2017, von https://medium.com/ionic-and-the-mobile-web/ionics-2016-by-the-numbers-3a8e2c65177b

Malavolta, I. (2016). Beyond native apps: web technologies to the rescue! (S. 1–2). Gehalten auf der Proceedings of the 1st International Workshop on Mobile Development, ACM. https://doi.org/10.1145/3001854.3001863

McCade, S. (2017, September 25). Native apps vs Web apps: A designer's perspective. Abgerufen 11. Januar 2018, von https://medium.muz.li/native-apps-vs-web-apps-a-designers-perspective-f44429d53e06

Meier, R. (2014). *Android App-Entwicklung: Die Gebrauchsanleitung für Programmierer*. John Wiley & Sons.

Meteor Development Group. (2017). Build Apps with JavaScript | Meteor. Abgerufen 6. Dezember 2017, von https://www.meteor.com/

Microsoft. (2017). Windows Dev Center. Abgerufen 30. November 2017, von https://developer.microsoft.com/de-de/windows

Milliman, T. (2017, Mai 11). After building my first React Native app, I'm now convinced it's the future. Abgerufen 30. November 2017, von https://medium.freecodecamp.org/after-building-my-first-react-native-app-im-now-convinced-it-s-the-future-d3c5e74f8fa8

Mohammadi Kho'i, F., & Jahid, J. (2016). *Comparing Native and Hybrid Applications with focus on Features*.

MongoDB and MySQL Compared. (2017). Abgerufen 3. Januar 2018, von https://www.mongodb.com/compare/mongodb-mysql

Neary, A. (2017, Mai 15). Rearchitecting Airbnb's Frontend. Abgerufen 13. Dezember 2017, von https://medium.com/airbnb-engineering/rearchitecting-airbnbs-frontend-5e213efc24d2

Neuhaus, J. (2017, August 28). Angular vs. React vs. Vue: A 2017 comparison. Abgerufen 10. Januar 2018, von https://medium.com/unicorn-supplies/angular-vs-react-vs-vue-a-2017-comparison-c5c52d620176

Nielsen, J. (1992). Finding Usability Problems Through Heuristic Evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (S. 373–380). New York, NY, USA: ACM. https://doi.org/10.1145/142750.142834

npm Inc. (2017). npm. Abgerufen 4. Januar 2018, von https://www.npmjs.com/features

Osmani, A. (2015, Dezember 23). AddyOsmani.com - Getting started with Progressive Web Apps. Abgerufen 13. Dezember 2017, von https://addyosmani.com/blog/getting-started-with-progressive-web-apps/

Österreichische Bundesbahnen. (1996, August). Richtline für die Überwachung und Inspektion von Eisenbahnbrücken in Spannbetonweise.

Park, H., Kim, K., & Lee, K. (2016). Geo-data visualization on online and offline mode of mobile web using HTML5. In *2016 4th International Workshop on Earth Observation and Remote Sensing Applications (EORSA)* (S. 237–240). https://doi.org/10.1109/EORSA.2016.7552804

Parker, Z., Poe, S., & Vrbsky, S. V. (2013). Comparing NoSQL MongoDB to an SQL DB. In *Proceedings of the 51st ACM Southeast Conference* (S. 5:1–5:6). New York, NY, USA:

ACM. https://doi.org/10.1145/2498328.2500047

Passaglia, A. (2017). *Vue.js 2 Cookbook*. Packt Publishing Ltd.

PHP Group. (2017). PHP: Was ist PHP? - Manual. Abgerufen 15. Januar 2018, von http://php.net/manual/de/intro-whatis.php

Ross, M. (2013). Cross-Plattform-Apps mit PhoneGap entwickeln. Abgerufen 2. Dezember 2017, von https://www.heise.de/developer/artikel/Cross-Plattform-Apps-mit-PhoneGap-entwickeln-1934535.html

Schalungsanker. (2017, November 8). In *Wikipedia*. Abgerufen von https://de.wikipedia.org/w/index.php?title=Schalungsanker&oldid=170787530

Sencha Inc. (2015, Februar 26). Ext JS + jQTouch + Raphaël = Sencha | Blog. Abgerufen 7. Dezember 2017, von https://www.sencha.com/blog/

Shneiderman, B. (1987). Shneiderman's Eight Golden Rules of Interface Design. Abgerufen von https://faculty.washington.edu/jtenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html

Sillmann, T. (2017). *Apps für iOS 10 professionell entwickeln: Sauberen Code schreiben mit Swift 3 und Objective-C. Stabile Apps für iPhone und iPad programmieren. Techniken & Methoden von Grund auf verstehen*. Carl Hanser Verlag GmbH Co KG.

Statista. (2017a). Brücken- und Tunnelbau in Österreich - Umsatz bis 2020 | Prognose. Abgerufen 18. November 2017, von https://ezproxy.fhstp.ac.at:2081/prognosen/403980/bruecken--und-tunnelbau-in-oesterreich---umsatzprognose

Statista. (2017b, Oktober). Österreich - Genutzte mobile Betriebssysteme 2017 | Statistik. Abgerufen 30. November 2017, von https://ezproxy.fhstp.ac.at:2081/statistik/daten/studie/303829/umfrage/genutzte-mobile-betriebssysteme-in-oesterreich/

Viennot, N., Garcia, E., & Nieh, J. (2014). A measurement study of google play (S. 221–233). ACM Press. https://doi.org/10.1145/2591971.2592003

Viering, R. (2017, Februar 24). Single-Page-Anwendungen Framework-unabhängig entwickeln. Abgerufen 4. Januar 2018, von https://www.heise.de/developer/artikel/Single-Page-Anwendungen-Framework-unabhaengig-entwickeln-3633930.html

Vogelsteller, F., Strack, I., & Reyna, M. (2016). *Meteor: Full-Stack Web Application Development*. Packt Publishing Ltd.

WPM Ingenieure GmbH. (2013). SIB-BAUWERKE.

WPM Ingenieure GmbH. (2018). Abgerufen 1. Januar 2018, von http://www.wpm-ingenieure.de/include.php?path=sib_bauwerke&pid=21&subid=35

Xanthopoulos, S., & Xinogalos, S. (2013). A comparative analysis of cross-platform development approaches for mobile applications (S. 213–220). Gehalten auf der Proceedings of the 6th Balkan Conference in Informatics, ACM. https://doi.org/10.1145/2490257.2490292

Zeller, B. (2016, August 12). *Unterstützung des gastronomischen Serviceprozesses durch vernetzte Mobilgeräte bei Veranstaltungen*. FH St.Pölten, St.Pölten.

Zhang, J., Liang, H., Chen, F., & Li, Y.-Q. (2016). Research and application of bridge inspection and maintenance management system. In *Software Engineering and Service Science (ICSESS), 2016 7th IEEE International Conference on* (S. 273–276). IEEE.

Abbildungsverzeichnis

Abbildung 1: System Module der Applikation Inspection Management Subsystem und Maintenance Management Subsystem (Zhang u. a., 2016) 16

Abbildung 2: Ablaufdiagramm des Reporting Moduls der Applikation Inspection Management Subsystem und Maintenance Management Subsystem (Zhang u. a., 2016) 17

Abbildung 3: Ablaufdiagramm des Wartungssystems der Applikation Inspection Management Subsystem und Maintenance Management Subsystem (Zhang u. a., 2016) 18

Abbildung 4: Anwendungsscreen von SIB-Bauwerke bei Erstellung von Einzelschäden 21

Abbildung 5: Architektur einer Web Applikation („Easyrec“, 2017) 23

Abbildung 6: Ablaufdiagramm einer Meteor Applikation (Meteor Development Group, 2017) 25

Abbildung 7: Auswertung von Sencha Touch/Ext JS durch Google Trends 27

Abbildung 8: Vergleich der Performance bei Zugriff auf den Speicher (Neuhaus, 2017) 29

Abbildung 9: Benchmark Tests der JS Bibliotheken (Neuhaus, 2017)..... 30

Abbildung 10: Darstellung der PWA von Pokedex.org (Lawson, 2015) 31

Abbildung 11: Bewertung von Hybrid und Nativen Apps (Ali & Mesbah, 2016) .. 35

Abbildung 12: Distributionskategorien von Adobe Air Applikationen (Ali & Mesbah, 2016) 37

Abbildung 13: iOS Systemarchitektur (Sillmann, 2017) 40

Abbildung 14: iOS Guidelines für die Navigationsleiste (Apple, 2018)..... 42

Abbildung 15: Android Guidelines für die Navigationsleiste (Google, 2018)..... 42

Abbildung 16: Checkliste für die Schadenskategorie F (Österreichische Bundesbahnen, 1996) 58

Abbildung 17: Der erste Teil der Hilfstabelle der möglichen Rissarten (Österreichische Bundesbahnen, 1996)..... 60

Abbildung 18: Hüllrohr ist das Symbol 5 in der Symbolgrafik („Schalungsanker“, 2017) 61

Abbildung 19: Beispiel Zustandsbewertung anhand einer Schadenstabelle (Österreichische Bundesbahnen, 1996) 62

Abbildung 20: Beispielbild eines Prüfberichtes (Bundesministerium für Verkehr, Bau und Stadtentwicklung, 2013) 65

Abbildung 21: Revisionsblatt bei Brückeninspektionen 67

Abbildung 22: Revisionsblatt Brückeninformationen 67

Abbildung 23: Revisionsblatt Veranlassung 68

Abbildung 24: Beispiel für eine Veranlassung „Sonstiges“ 68

Abbildung 25: Fotomaterial einer Brückeninspektion 69

Abbildung 26: Systemarchitektur der Applikation 77

Abbildung 27: Funktionsabbildung einer API Schnittstelle 80

Abbildung 28: Datenbankstruktur der Applikation 85

Abbildung 29: Abbildung von dem Screen "Brücke hinzufügen" 89

Abbildung 30: Brückenübersicht der Applikation 90

Abbildung 31: Inspektionsübersicht der Applikation 91

Abbildung 32: Darstellung der Funktionalität „Hochladen von Fotomaterial“ 92

Abbildung 33: Setzen einer Veranlassung in der Applikation 93

Abbildung 34: Screen der Schadensmarkierung in der Applikation 94

Tabellenverzeichnis

Tabelle 1: JS Frontend Bibliotheken im Vergleich (Stand 10.01.2018) 28

Tabelle 2: Auswertung der Marktführer von CPT (Ali & Mesbah, 2016) 34

Tabelle 3: Direkter Vergleich der drei Entwicklungsoptionen 44

Tabelle 4: Performance Test von nativen und hybriden Applikationen (Corral, Sillitti, & Succi, 2012) 45

Tabelle 5: Performance Test zwischen MySQL und MongoDB (Caetano, 2015/2017) 78

Tabelle 6: Demographische Daten der Testpersonen 97

Tabelle 7: Ergebnisse der aufgabenbasierten Tests 98

Listingverzeichnis

Listing 1: Composer Befehl zur Erstellung eines neuen Laravel Projektes 81

Listing 2: Vue Ressourcen wurden zum Projekt hinzugefügt 82

Listing 3: Konfiguration des Build Systems 82

Listing 4: Laravel Routing 83

Listing 5: Vue Routing bzw. URL Aufbau der Applikation 85

Listing 6: API Routing 86

Listing 7: Definition einer Textbox mit Fehlermeldung 87

Listing 8: Definition einer Textbox mit Fehlermeldung 87

Listing 9: Speicherung einer Brücke in die Datenbank 88

Anhang

Experten und Expertinnen Fragen und Testaufgaben

Einführung

Herzlich Willkommen und danke, dass Sie sich für den Usertest Zeit nehmen.

Mein Name ist Gerald Strobl und ich schreibe derzeit meine Diplomarbeit und untersuche, wie der Prozess bei Brückeninspektionen durch mobile Entwicklungsmethoden verbessert werden kann. Dazu soll eine mobile Anwendung entstehen, die vom Fachpersonal für Brückeninspektionen eingesetzt wird.

Ziel dieser Evaluierung ist es, die Idee und Benutzbarkeit der Applikation zu testen. Wichtig dabei ist, dass Sie als Testperson keinen Fehler machen können. Haben Sie während des Tests Fragen oder Anmerkungen, zögern Sie nicht, diese zu stellen oder auszusprechen. Für die Evaluierung haben ehrliche Bemerkungen einen großen Mehrwert.

Vom Interview wird eine Audioaufnahme und vom Testgerät ein Screenrecording gemacht. Die Angaben werden vertraulich behandelt, anonymisiert ausgewertet und im Rahmen der Diplomarbeit veröffentlicht. Die Rohdaten werden nicht an Dritte weitergeben oder veröffentlicht.

Am Beginn werde ich Ihnen ein paar allgemeine Fragen zu Ihrer Person und Ihrer Erfahrung im Bereich der Brückeninspektion stellen. In weiterer Folge bekommen Sie Aufgaben zum Lösen, diese behandeln einzelne Teilprozesse einer Brückeninspektion. Zum Abschluss werden noch Fragen bezüglich der Handhabung und Benutzerbarkeit der Applikation gestellt.

Hiermit erkläre ich _____, dass ich mit den oben genannten Bedingungen, das gesammelte Datenmaterial für wissenschaftliche Zwecke im Rahmen der Diplomarbeit von Gerald Strobl anonymisiert zu verwenden, einverstanden bin.

Datum:_____ Unterschrift: _____

Allgemeine Fragen zur Person

Demographische Daten

Alter: _____

Geschlecht: □ Weiblich □ Männlich

Beruf: _____

Ausbildung: _____

Mediennutzung

Smartphone/Tablet: □ Ja □ Nein

Gerät: _____

Betriebssystem: _____

Browser: _____

Erfahrung Brückeninspektionen

Kennen Sie den Ablauf einer Brückeninspektion: □ Ja □ Nein

In welchem Berufsfeld hatten Sie bereits tun mit Brückeninspektionen: _____

Aufgabe 1

Auf der Strecke Wien-Krems muss eine Inspektion durchgeführt werden. Bitte legen Sie auf dieser Strecke eine Brücke an. Die Brücke liegt auf km-Stelle 25 und ist eine Stahlbetonbrücke. Sie wollen nur die Brücke anlegen, die Inspektionseinträge werden später hinzugefügt.

Aufgabenstellung hat einen Praxisbezug: □ Ja □ Nein

Handhabung der Funktionalität:

□ Sehr Gut □ Gut □ Befriedigend □ Genügend □ Nicht Genügend

Aufgabe 2

Sie haben falsche Angaben vom Vorgesetzten bekommen. Die km-Stelle liegt bei 21. Bitte ändern Sie diese Eigenschaften der Brücke.

Aufgabenstellung hat einen Praxisbezug: □ Ja □ Nein

Handhabung der Funktionalität:

□ Sehr Gut □ Gut □ Befriedigend □ Genügend □ Nicht Genügend

Aufgabe 3

Sie führen jetzt eine Inspektion durch und haben zwei Bilder von den Bauobjekten mit Ihrem Tablet gemacht. Laden Sie die Bilder auf die Plattform hoch. Zusätzlich wollen Sie folgende Schäden nur auf dem **ersten** Bild markieren:

1. Zwei vertikale Risse über dem linken Träger. Klassifizierung A
2. Ein vertikaler Riss über die volle Tragwerkshöhe Pfeiler C L D B. Klassifizierung A

Aufgabenstellung hat einen Praxisbezug: □ Ja □ Nein

Handhabung der Funktionalität:

□ Sehr Gut □ Gut □ Befriedigend □ Genügend □ Nicht Genügend

Aufgabe 4

Bitte markieren Sie auf dem zweiten Bild folgenden Schaden:

1. Korrosionsschäden am Geländer. Klassifizierung K

Aufgabenstellung hat einen Praxisbezug: □ Ja □ Nein

Handhabung der Funktionalität:

□ Sehr Gut □ Gut □ Befriedigend □ Genügend □ Nicht Genügend

Aufgabe 5

Sie müssen eine Veranlassung treffen, weil das Steinschlaggitter beim Bauobjekt fehlt, die Untersuchungsfrist muss verkürzt werden. Das Ausführungsdatum auf 20.03.2018 ändern.

Aufgabenstellung hat einen Praxisbezug: □ Ja □ Nein

Handhabung der Funktionalität:

□ Sehr Gut □ Gut □ Befriedigend □ Genügend □ Nicht Genügend

Aufgabe 6

Eine Inspektion aus dem Jahr 2010 muss für die Strecke Wien-Mistelbach nachgetragen werden. Machen Sie ein Bild mit dem Tablet und fügen Sie folgende Daten hinzu:

2. Inspektionsdatum 20.05.2010
3. Schäden am Bild markieren: An den Stirnwänden fließt Dichtungsmaterial heraus. Klassifizierung H

Aufgabenstellung hat einen Praxisbezug: □ Ja □ Nein

Handhabung der Funktionalität:

□ Sehr Gut □ Gut □ Befriedigend □ Genügend □ Nicht Genügend

Aufgabe 7

Löschen Sie die Inspektion aus dem Jahr 2011 auf der Strecke Wolkersdorf-Leopoldau.

Aufgabenstellung hat einen Praxisbezug: □ Ja □ Nein

Handhabung der Funktionalität:

□ Sehr Gut □ Gut □ Befriedigend □ Genügend □ Nicht Genügend

Aufgabe 8

Bearbeiten Sie die Inspektion am 2017-12-19 auf der Strecke Wien-Krems an der km-Stelle 35. Ziel ist es, die Markierung „Testschaden“ aus dem ersten Bild der Inspektion zu löschen.

Aufgabenstellung hat einen Praxisbezug: □ Ja □ Nein

Handhabung der Funktionalität:

□ Sehr Gut □ Gut □ Befriedigend □ Genügend □ Nicht Genügend

Aufgabe 9

Es wurden fehlerhafte Angaben zur Inspektion gemacht, bitte ändern Sie die Inspektionsdaten bei Wien-Krems km-Stelle 35 vom 08.11.2017 und setzen Sie die Veranlassung auf Beobachtung.

Aufgabenstellung hat einen Praxisbezug: □ Ja □ Nein

Handhabung der Funktionalität:

□ Sehr Gut □ Gut □ Befriedigend □ Genügend □ Nicht Genügend

Aufgabe 10

Ihr Vorgesetzter möchte den kompletten Inspektionsverlauf von der Brücke Wien-Krems km-Stelle 35 exportiert haben und zusätzlich die Inspektion aus dem Jahr 2006.

Aufgabenstellung hat einen Praxisbezug: □ Ja □ Nein

Handhabung der Funktionalität:

□ Sehr Gut □ Gut □ Befriedigend □ Genügend □ Nicht Genügend

Spezifische Fragen

1. Wie ist Ihr allgemeiner Eindruck von der Applikation? Ist die Applikation für eine reale Brückeninspektion verwendbar?

2. Haben Sie bei der Bedienung und Handhabung der Applikation eine Funktionalität vermisst?

3. Gibt es eine Funktion, die Sie sich anders vorgestellt bzw. haben Sie Ideen, wie die Funktion anders abgebildet werden kann?

4. Würde diese Applikation (in einer weiterentwickelten Version) Potential für den praktischen Einsatz bei einer Brückeninspektion bieten?

5. Sonstige Anmerkungen oder Verbesserungsvorschläge?