

Konzertkalender

Damian Senn

TSBE

DIPLOMARBEIT

Konzertkalender

Author:
Damian Senn

Experten:
Sandro Bertolino
Severin Rätz

Eine Diplomarbeit für den Abschluss Dipl. Techniker Informatik

13. April 2019

Authentizität

Mit meiner Unterschrift bestätige ich, die vorliegende Diplomarbeit selbstständig, ohne Hilfe Dritter und nur unter Benutzung der angegebenen Quellen ohne Copyright-Verletzung, erstellt zu haben.

Unterschrift:

Ort:

Datum:

TSBE

Management Summary

Dipl. Techniker Informatik

Konzertkalender

von Damian Senn

Inhalt für Management Summary folgt hier...

Danksagung

TODO

Inhaltsverzeichnis

Authentizität	i
Management Summary	ii
Danksagung	iii
Abbildungsverzeichnis	vii
Tabellenverzeichnis	viii
Abkürzungsverzeichnis	ix
1 Initialisierung	1
1.1 Ausgangslage	1
1.2 Projektziele	2
1.3 Projektorganisation	3
1.4 Ausgefüllter Projektplan	4
1.5 Lieferergebnisse	5
1.6 Ressourcenplan	5
1.7 Risiken	5
1.8 Abgrenzungen	6
1.9 Studie	7
1.9.1 Informationsbeschaffung	7
1.9.2 Anforderungskatalog	8
1.9.3 Pflichtenheft	9
1.9.4 Mögliche Varianten	9
1.9.5 Evaluation Varianten	9
1.9.6 Entscheid Varianten	9
1.9.7 Wirtschaftlichkeit	9
2 Konzept	10
2.1 Design	10
2.2 Software	10
2.3 Testing	10
3 Realisierung	11
3.1 Umsetzung	11
3.2 Tests	11
3.3 Auswertung	11
4 Einführung	12
4.1 Projektcontrolling	12
4.2 Wirtschaftlichkeit	12

5	Schlussbetrachtung	13
A	Projektinitialisierungsauftrag	14
B	Projektauftrag	18
B.1	Ausgangslage	18
B.2	Projektziele	19
B.3	Terminplan	20
B.4	Meilensteine	20
B.5	Organigramm	21
B.5.1	Tätigkeiten im Projekt	21
B.5.2	Kommunikation	21
B.6	Abgrenzungen	22
B.7	Anforderungskatalog	23
B.8	Lösungsbeschreibung	25
B.9	Risiken	26
B.9.1	Projektrisiken	27
B.9.2	Massnahmen	28
B.9.3	Risikodiagramm ohne Massnahmen	29
B.9.4	Risikodiagramm mit Massnahmen	30
C	Terminplan	31
D	Studie	32
D.1	Zweck des Dokuments	32
D.2	Informationsbeschaffung	32
D.3	Anforderungskatalog	33
D.4	Evaluation Browser-Technologie	35
D.4.1	Variante: React	35
D.4.2	Variante: Next.js	35
D.4.3	Variante: SSR	35
D.5	Bewertungen Browser-Technologie	36
D.6	Entscheid Browser-Technologie	36
D.7	Evaluation Server-Technologie	37
D.7.1	Variante: Node.js / koa.js	37
D.7.2	Variante: Elixir / Phoenix	37
D.7.3	Variante: Next.js	37
D.8	Bewertungen Server-Technologie	38
D.9	Entscheid Server-Technologie	38
D.10	Evaluation Testing-Technologie	39
D.10.1	Jest + Puppeteer	39
D.10.2	Wallaby	39
D.11	Bewertungen Testing-Technologie	40
D.12	Entscheid Testing-Technologie	40
D.13	Wirtschaftlichkeit	41
D.13.1	Break Even Analyse	41
E	Konzept	42
E.1	Design- und Bedienkonzept	42
E.2	Softwarekonzept	42
E.3	Testkonzept	42

F	Arbeitsjournal	43
F.1	Sonntag 3. März	43
F.2	Dienstag 5. März	43
F.3	Mittwoch 6. März	43
F.4	Samstag 9. März	43
F.5	Dienstag 12. März	43
F.6	Samstag 16. März	44
F.7	Dienstag 19. März	44
F.8	Mittwoch 27. März	44
F.9	Sonntag 31. März	44
F.10	Sonntag 31. März	44
F.11	Freitag 5. April	44
F.12	Samstag 6. April	44
F.13	Mittwoch 10. April	45

Abbildungsverzeichnis

1.1	Organigram	3
1.2	Abgrenzungen	6
B.1	Organigram	21
B.2	Abgrenzungen	22
B.3	Phoenix Framework	25
B.4	Wallaby	25

Tabellenverzeichnis

1.1	Ziele	2
1.2	Anforderungskatalog	9
B.1	Ziele	19
B.2	Terminplan	20
B.3	Meilensteine	20
B.4	Tätigkeiten Verteilung	21
B.5	Anforderungskatalog	24
B.6	Risiken - Schadensskala	26
B.7	Risiken - Eintrittswahrscheinlichkeit	26
B.8	Risiken - Handlungen zur Senkung der Bewertung	26
B.9	Projektrisiken	27
B.10	Projektrisiken	28
D.1	Informationsbeschaffung	32
D.2	Anforderungskatalog	34
D.3	Browser-Technologie Kriterien	35
D.4	Browser-Technologie Bewertung	36
D.5	Server-Technologie Kriterien	37
D.6	Server-Technologie Bewertung	38
D.7	Testing-Technologie Kriterien	39
D.8	Testing-Technologie Bewertung	40

Abkürzungsverzeichnis

HTML	H ypertext M arkup L anguage
CSS	C ascading S yle S heets
SEO	S earch E ngine O ptimization
OWASP	O pen W eb A pplication S ecurity P roject
XSS	C ross-site scripting
SSR	S erver S ide R endered

For/Dedicated to/To my...

Kapitel 1

Initialisierung

1.1 Ausgangslage

1.2 Projektziele

Nr.	Zielbeschreibung	Muss/Kann
Produktziele		
1.1	Besucher können im Produkt nach Konzerten suchen	Muss
1.2	Suchresultate können nach Musik-Genre und Ort gefiltert werden	Muss
1.3	Besucher können Details zu einem Konzert ansehen	Muss
1.4	Das Produkt soll ein modernes responsives Design vorweisen	Muss
1.5	Konzerte sollen von Suchmaschinen indexiert werden können	Muss
1.6	Benutzer können sich im Produkt registrieren	Muss
1.7	Benutzer können ihr Passwort nach Verlust neu setzen	Muss
1.8	Inhalte des Portals sind durch die Benutzer erfassbar und bearbeitbar	Muss
1.9	Kompatibilität mit aktuellem Google Chrome und Mozilla Firefox Browser	Muss
1.10	Konzerte können vom Produkt nach Facebook exportiert werden	Kann
1.11	Ein angemeldeter Benutzer kann vermerken ob er einem Konzert teilnimmt	Kann
1.12	Das Produkt soll sich an die Security Best-Practices von OWASP halten	Muss
Abwicklungsziele		
2.1	Das Projekt soll nach HERMES 5 unter Berücksichtigung der Richtlinien von der TSBE dokumentiert werden	Muss
2.2	Das Produkt muss bis Projektende fertiggestellt, getestet und bereit für die Einführung sein	Muss
2.3	Die Technische-Umsetzung wird durch Damian Senn erstellt	Muss
2.4	Die Kommunikation zwischen Experten und Diplomanden erfolgt wie im Projektauftrag B.5.2 beschrieben.	Muss
2.5	Das Projekt muss bis Ende Mai 2019 abgeschlossen sein	Muss

TABELLE 1.1: Ziele

1.3 Projektorganisation

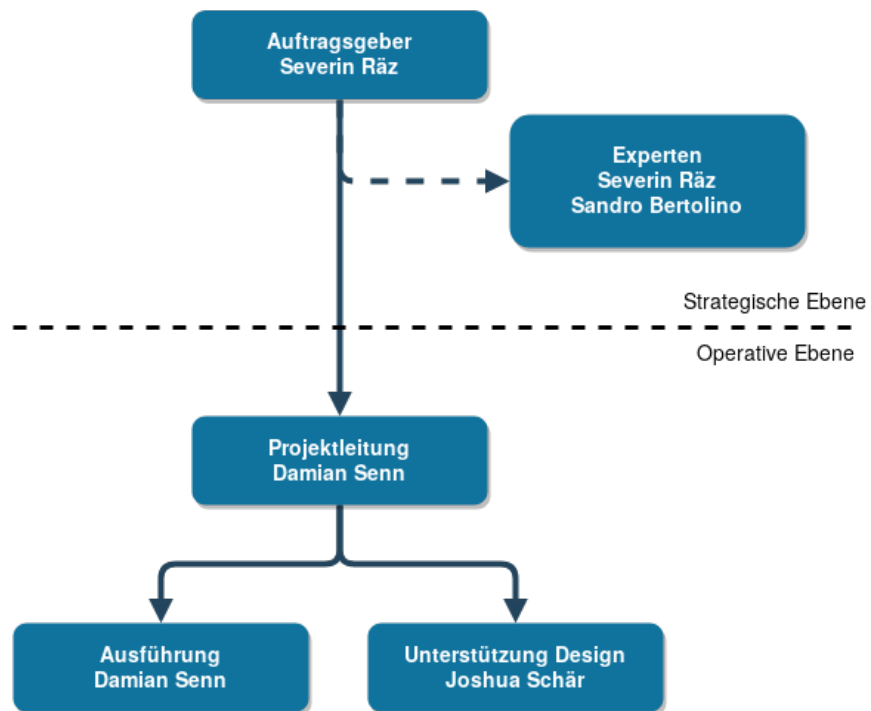


ABBILDUNG 1.1: Organigramm

1.4 Ausgefüllter Projektplan

Projektplan: Konzertkalender

[illegible]

Total Soll:	286
Bereits benötigt:	49
Restliche Stunden:	237

Legende	
Name	Abk.
Damian Senn	DS
Sandro Bertolino	SB
Severin Rätz	SR
Joshua Schär	JS

1.5 Lieferergebnisse

1.6 Ressourcenplan

1.7 Risiken

1.8 Abgrenzungen

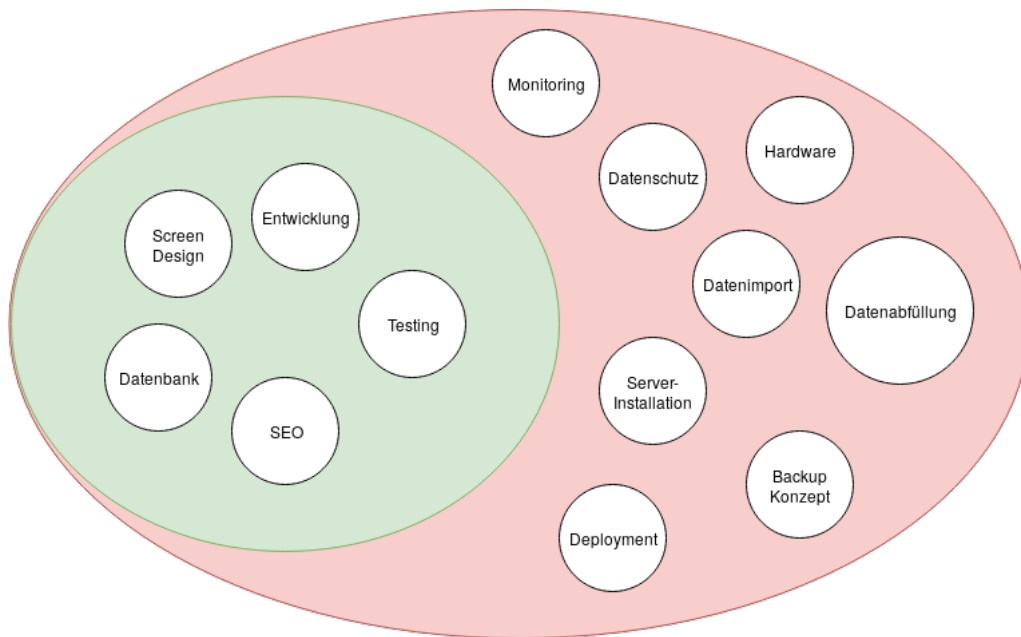


ABBILDUNG 1.2: Abgrenzungen

Die detaillierten Erklärung zu den Abgrenzungen sind im Projektauftrag **B.6** zu finden.

1.9 Studie

1.9.1 Informationsbeschaffung

1.9.2 Anforderungskatalog

Feature	Titel	Nr.	Kriterium	Ziel	Muss
Suche	Suche nach Konzertname	1.1	Listet alle Konzerte die Wörter der Suche im Konzertnamen beinhalten	1.1	Muss
	Suche nach Konzertlocation	1.2	Schränkt die Such-Resultate nach gegebener Konzertlocation ein	1.2	Muss
	Suche nach Ort	1.2	Schränkt die Such-Resultate nach gegebenem Ort ein	1.2	Muss
	Suche nach Genre	1.2	Schränkt die Such-Resultate nach gegebenem Musik-Genre ein	1.2	Muss
Design	Desktop	2.1	Alle Ansichten haben eine Desktop-Optimierte Variante	1.4	Muss
	Tablet	2.2	Alle Ansichten haben eine Tablet-Optimierte Variante	1.4	Muss
	Mobile	2.3	Alle Ansichten haben eine Mobile-Optimierte Variante	1.4	Muss
	Browser Kompatibilität	2.4	Alle Ansichten müssen in aktuellem Google Chrome und Mozilla Firefox dem Grundlayout folgen	1.9	Muss
SEO	Indexierbarkeit	3.1	Das Produkt ist von Suchmaschinen indexierbar	1.5	Muss
	Linked Data	3.2	Konzert Detailseiten sind mit dem Event-Schema ¹ ausgestattet	1.5	Muss
Benutzer	Registrierung	4.1	Besucher können sich einen Benutzer registrieren, Benutzernamen und E-Mail Adressen müssen einzigartig sein	1.6	Muss
	Passwort-Vergessen	4.2	Benutzer können sich einen Passwort-Reset Link anfordern	1.7	Muss
	Social	4.3	Benutzer können auf Konzerten vermerken ob sie teilnehmen oder nicht	1.11	Kann

¹<https://schema.org/Event>

Feature	Titel	Nr.	Kriterium	Ziel	Muss
Erfassung	Artist	5.1	Benutzer können Artisten mit einem Genre erfassen	1.8	Muss
	Location	5.2	Benutzer können eine Konzertlocation mit Ort/Strasse erfassen	1.8	Muss
	Konzert	5.3	Benutzer können ein Konzert mit Konzertlocation und Artisten erfassen	1.8	Muss
	Facebook	5.4	Benutzer können ein Konzert in ein Facebook-Event exportieren	1.10	Kann
Security	SQL-Injection	6.1	Das Produkt soll resistent gegen SQL-Injection sein	1.12	Muss
	HTML-Injection	6.2	Das Produkt soll resistent gegen HTML-Injection / XSS sein	1.12	Muss
	Passwort encryption	6.3	Passwörter von Benutzer müssen mit einem sicheren Verfahren gespeichert werden	1.12	Muss
	Session	6.4	Session-Cookies dürfen nicht durch JavaScript ausgelesen werden	1.12	Kann
Performance	Ladezeit	7.1	Die Seitenansichten dürfen nicht länger als 6 Sekunden auf einem 3G Netz laden		Muss

TABELLE 1.2: Anforderungskatalog

1.9.3 Pflichtenheft**1.9.4 Mögliche Varianten****1.9.5 Evaluation Varianten****1.9.6 Entscheid Varianten****1.9.7 Wirtschaftlichkeit**

Kapitel 2

Konzept

2.1 Design

2.2 Software

2.3 Testing

Kapitel 3

Realisierung

3.1 Umsetzung

3.2 Tests

3.3 Auswertung

Kapitel 4

Einführung

4.1 Projektcontrolling

4.2 Wirtschaftlichkeit

Kapitel 5

Schlussbetrachtung

PROJEKTINITIALISIERUNGSAUFTTRAG

WEBBASIERTER KONZERTKALENDER

Auftraggeber: Damian Senn

Projektleiter: Damian Senn

Autor: Damian Senn

1	Ausgangslage	2
2	Ziele	3
3	Rahmenbedingungen	3
4	Ergebnisse und Termine	3
5	Aufwand	3
6	Kosten	4
7	Ressourcen	4
8	Kommunikation	4
9	Risiken	4

AUSGANGSLAGE

Als regelmässiger Konzertbesucher wünsche ich mir eine Plattform im Internet, auf welcher ich eine zuverlässige Übersicht an Konzerten in meiner Umgebung vorfinde. Heute sind die Events nur verteilt auf verschiedenen Seiten wie die der Venues, des Konzertveranstalters, des Künstlers oder auf Facebook publiziert.

Ich möchte deshalb eine zentrale Plattform entwickeln, die es Benutzern einfach macht, Konzerte für ihren Geschmack zu finden.

Die Plattform soll Genre unabhängig sein und entsprechende Filter anbieten.

Um einen zusätzlichen Service für den User zur Verfügung zu stellen, ist es auch denkbar, eine Art Notifikationssystem zu bauen um Benutzer über Handy-Notifications oder per Email an Konzerte oder Künstler zu erinnern.

Konzertveranstaltern kann das Erfassen ihrer Events vereinfacht werden, indem auf der Plattform erfasste Veranstaltungen direkt auf den Sozialen Medien wie Facebook, Twitter oder Instagram geteilt werden können.

ZIELE

- Definition der funktionalen Anforderungen
- Definition der nicht funktionalen Anforderungen
- Definition Projektumfang
- Projektplanung
- Aufwandschätzung
- Technologie Evaluierungen
- Lösungsvarianten

RAHMENBEDINGUNGEN

- Das Projekt wird im Rahmen der Diplomarbeit durchgeführt
- Richtlinien zum Erstellen des Diplomberichtes
- Anwendung von HERMES, angepasst auf das Projekt

ERGEBNISSE UND TERMINE

- Studie
- Projektauftrag
- Projektplan
- Evaluation
- Festgelegter Scope

AUFWAND

Der Aufwand der Diplomarbeit wird auf ca. 300 Stunden geschätzt. Für die Initialisierungsphase wird mit ca. einer Woche gerechnet.

Initialisierung: 42h

KOSTEN

Die Kosten werden mit einem durchschnittlichen Stundensatz von CHF 150.– gerechnet:

Initialisierung: CHF 6300.–

RESSOURCEN

Personal

Damian Senn (ca. 300 Stunden)

Da das Projekt durch Damian Senn alleine durchgeführt wird, ist keine Ressourcen aufteilung nötig.

Sachmittel

Es werden keine Sachmittel wie Räume, IT-Infrastruktur, Spezifische Software, etc. benötigt die externe Kosten verursachen.

KOMMUNIKATION

Da das Projekt von Damian Senn alleine durchgeführt wird, gibt es keine zu definierende Kommunikationswege.

RISIKEN

Es sind keine Risiken für die Initialisierungsphase bekannt.

Anhang B

Projektauftrag

B.1 Ausgangslage

Als regelmässiger Konzertbesucher wünsche ich mir eine Plattform im Internet, auf welcher ich eine zuverlässige Übersicht an Konzerten in meiner Umgebung vorfinde. Heute sind die Events nur verteilt auf verschiedenen Seiten wie die der Venues, des Konzertveranstalters, des Künstlers oder auf Facebook publiziert.

Ich möchte deshalb eine zentrale Plattform entwickeln, die es Benutzern einfach macht, Konzerte für ihren Geschmack zu finden.

Die Plattform soll Genre unabhängig sein und entsprechende Filter anbieten.

Um einen zusätzlichen Service für den User zur Verfügung zu stellen, ist es auch denkbar, eine Art Notifikationssystem zu bauen um Benutzer über Handy-Notifications oder per Email an Konzerte oder Künstler zu erinnern.

Konzertveranstaltern kann das Erfassen ihrer Events vereinfacht werden, indem auf der Plattform erfasste Veranstaltungen direkt auf den Sozialen Medien wie Facebook, Twitter oder Instagram geteilt werden können.

B.2 Projektziele

Nr.	Zielbeschreibung	Muss/Kann
Produktziele		
1.1	Besucher können im Produkt nach Konzerten suchen	Muss
1.2	Suchresultate können nach Musik-Genre und Ort gefiltert werden	Muss
1.3	Besucher können Details zu einem Konzert ansehen	Muss
1.4	Das Produkt soll ein modernes responsives Design vorweisen	Muss
1.5	Konzerte sollen von Suchmaschinen indexiert werden können	Muss
1.6	Benutzer können sich im Produkt registrieren	Muss
1.7	Benutzer können ihr Passwort nach Verlust neu setzen	Muss
1.8	Inhalte des Portals sind durch die Benutzer erfassbar und bearbeitbar	Muss
1.9	Kompatibilität mit aktuellem Google Chrome und Mozilla Firefox Browser	Muss
1.10	Konzerte können vom Produkt nach Facebook exportiert werden	Kann
1.11	Ein angemeldeter Benutzer kann vermerken ob er einem Konzert teilnimmt	Kann
1.12	Das Produkt soll sich an die Security Best-Practices von OWASP halten	Muss
Abwicklungsziele		
2.1	Das Projekt soll nach HERMES 5 unter Berücksichtigung der Richtlinien von der TSBE dokumentiert werden	Muss
2.2	Das Produkt muss bis Projektende fertiggestellt, getestet und bereit für die Einführung sein	Muss
2.3	Die Technische-Umsetzung wird durch Damian Senn erstellt	Muss
2.4	Die Kommunikation zwischen Experten und Diplomanden erfolgt wie im Projektauftrag B.5.2 beschrieben.	Muss
2.5	Das Projekt muss bis Ende Mai 2019 abgeschlossen sein	Muss

TABELLE B.1: Ziele

B.3 Terminplan

Nachfolgend ist der grobe Terminplan für die geplanten Phasen. Im Anhang C ist der detaillierte Terminplan abgelegt.

Phase	Datum	Stunden
Initialisierung	06.03.2019 - 31.03.2019	64
Konzept	01.04.2019 - 14.04.2019	66
Realisierung	22.04.2019 - 19.05.2019	136
Abschluss	20.05.2019 - 26.05.2019	36
Total:		286

TABELLE B.2: Terminplan

B.4 Meilensteine

Nr.	Meilenstein	KW	Datum
1	Kickoff-Meeting	10	
2	Abschluss Phase Initialisierung	13	
3	Zwischen-Meeting	18	
4	Abschluss Phase Konzept	16	
5	Abschluss Phase Realisierung	20	
6	Abschluss Phase Einführung	21	
7	Abschluss-Meeting	22	

TABELLE B.3: Meilensteine

B.5 Organigramm

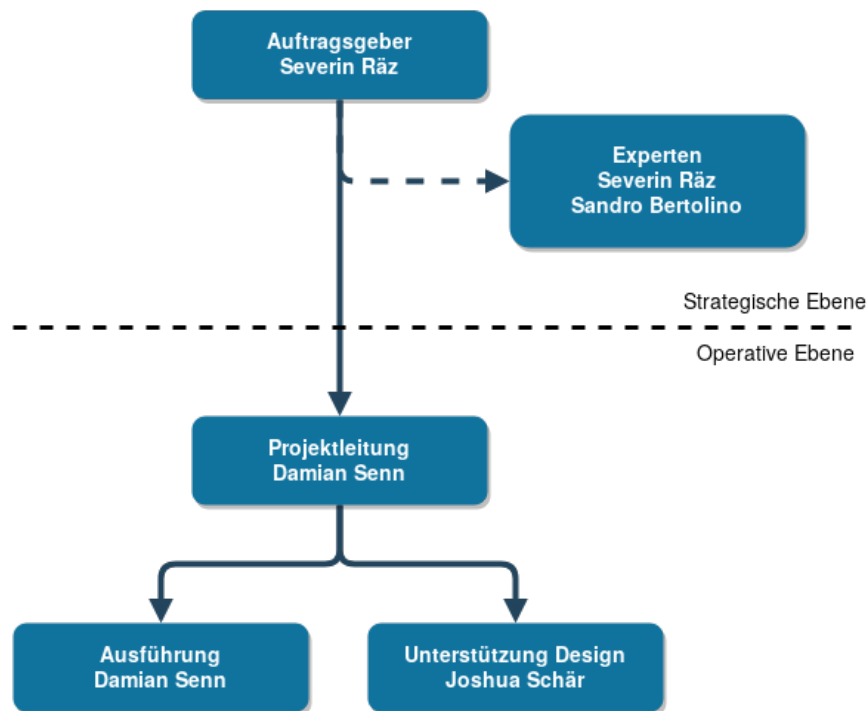


ABBILDUNG B.1: Organigramm

B.5.1 Tätigkeiten im Projekt

Für die Freigaben der Phasen ist nach Absprache mit Severin Rätz Damian Senn selbstständig verantwortlich.

Name	Funktions- und Tätigkeitsbereich
Severin Rätz	Auftraggeber, externer Experte
Sandro Bertolino	Interner Experte
Damian Senn	Projektleiter, Ausführung

TABELLE B.4: Tätigkeiten Verteilung

B.5.2 Kommunikation

Wie im Kickoff-Meeting besprochen, wird Damian Senn alle zwei Wochen einen kurzen Bericht an Sandro Bertolino und Severin Rätz per E-Mail schicken. Im Bericht wird erläutert, was in der Zwischenzeit erledigt wurde und was die nächsten Schritte im Projekt sind.

B.6 Abgrenzungen

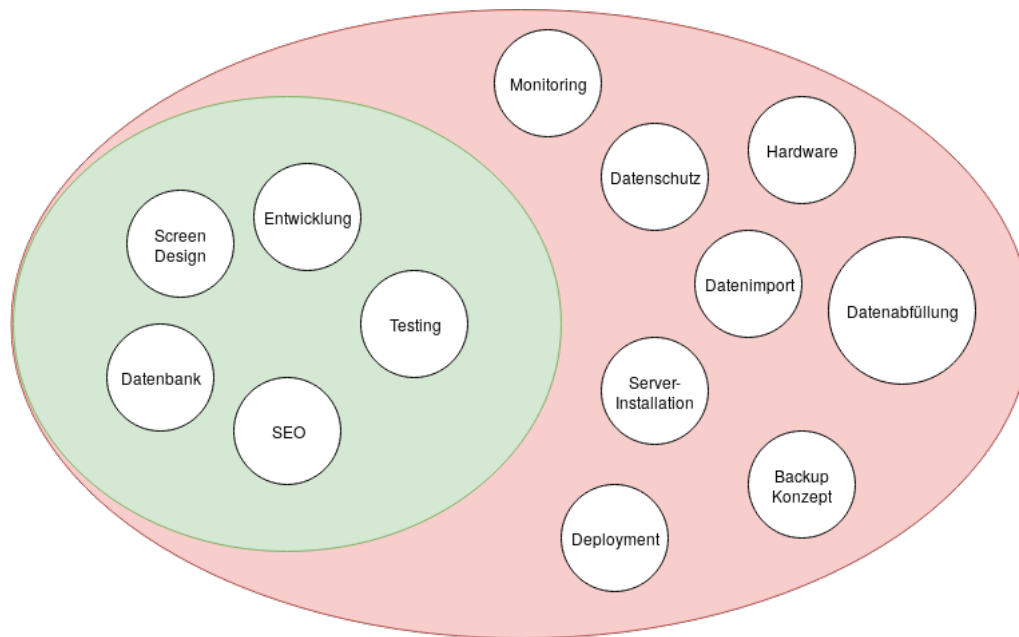


ABBILDUNG B.2: Abgrenzungen

Hardware, Server-Installation, Deployment und Monitoring

Da das Projekt ein reines Software-Entwicklungs Projekt ist, werden keine Operativen tätigkeiten wie Hardwarebeschaffung, Server-Installation, Deployment und das einrichten eines Monitoring-Systems vorgenommen.

Datenschutz

Da das Projekt nicht deployed wird und somit nicht produktiv /online gestellt wird, müssen im Rahmen dieser Projektarbeit noch keine Gedanken über den Datenschutz gemacht werden.

Datenimport

Da wir bisher keine existierenden Konzertdaten besitzen, ist es nicht nötig, einen Datenimport zu implementieren.

Datenabfüllung

Die Projektarbeit beinhaltet kein Datenset, Tests werden mit Testdaten abgewickelt. Es liegt nicht in der Verantwortung des Projektleiters, dass Daten in die Applikation abgefüllt werden.

Backup Konzept

Es wird kein Backup Konzept benötigt, da die Applikation im Rahmen dieses Projektes nicht produktiv geschaltet wird.

B.7 Anforderungskatalog

Der Anforderungskatalog wurde in der Studie erarbeitet. Es wurden Kann und Muss Kriterien definiert, wobei ein Muss-Kriterium zwingend erfüllt werden muss und ein Kann-Kriterium als Erweiterung angesehen wird.

Feature	Titel	Nr.	Kriterium	Ziel	Muss
Suche	Suche nach Konzertname	1.1	Listet alle Konzerte die Wörter der Suche im Konzertnamen beinhalten	1.1	Muss
	Suche nach Konzertlocation	1.2	Schränkt die Such-Resultate nach gegebener Konzertlocation ein	1.2	Muss
	Suche nach Ort	1.2	Schränkt die Such-Resultate nach gegebenem Ort ein	1.2	Muss
	Suche nach Genre	1.2	Schränkt die Such-Resultate nach gegebenem Musik-Genre ein	1.2	Muss
Design	Desktop	2.1	Alle Ansichten haben eine Desktop-Optimierte Variante	1.4	Muss
	Tablet	2.2	Alle Ansichten haben eine Tablet-Optimierte Variante	1.4	Muss
	Mobile	2.3	Alle Ansichten haben eine Mobile-Optimierte Variante	1.4	Muss
	Browser Kompatibilität	2.4	Alle Ansichten müssen in aktuellem Google Chrome und Mozilla Firefox dem Grundlayout folgen	1.9	Muss
SEO	Indexierbarkeit	3.1	Das Produkt ist von Suchmaschinen indexierbar	1.5	Muss
	Linked Data	3.2	Konzert Detailseiten sind mit dem Event-Schema ¹ ausgestattet	1.5	Muss
Benutzer	Registrierung	4.1	Besucher können sich einen Benutzer registrieren, Benutzernamen und E-Mail Adressen müssen einzigartig sein	1.6	Muss
	Passwort-Vergessen	4.2	Benutzer können sich einen Passwort-Reset Link anfordern	1.7	Muss
	Social	4.3	Benutzer können auf Konzerten vermerken ob sie Teilnehmen oder nicht	1.11	Kann

¹<https://schema.org/Event>

Feature	Titel	Nr.	Kriterium	Ziel	Muss
Erfassung	Artist	5.1	Benutzer können Artisten mit einem Genre erfassen	1.8	Muss
	Location	5.2	Benutzer können eine Konzertlocation mit Ort/Strasse erfassen	1.8	Muss
	Konzert	5.3	Benutzer können ein Konzert mit Konzertlocation und Artisten erfassen	1.8	Muss
	Facebook	5.4	Benutzer können ein Konzert in ein Facebook-Event exportieren	1.10	Kann
Security	SQL-Injection	6.1	Das Produkt soll resistent gegen SQL-Injection sein	1.12	Muss
	HTML-Injection	6.2	Das Produkt soll resistent gegen HTML-Injection / XSS sein	1.12	Muss
	Passwort encryption	6.3	Passwörter von Benutzer müssen mit einem sicheren Verfahren gespeichert werden	1.12	Muss
	Session	6.4	Session-Cookies dürfen nicht durch JavaScript ausgelesen werden	1.12	Kann
Performance	Ladezeit	7.1	Die Seitenansichten dürfen nicht länger als 6 Sekunden auf einem 3G Netz laden		Muss

TABELLE B.5: Anforderungskatalog

B.8 Lösungsbeschreibung

In der Studie (Anhang D) wurden Technologien gegenüber gestellt und für die Umsetzung mittels Nutzwertanalysen ausgewählt.

Folgende Technologien wurden ausgewählt:

Browser sowie Server Technologie:



ABBILDUNG B.3: Phoenix Framework

Quelle: <https://github.com/phoenixframework/phoenix>

Testing Technologie:



ABBILDUNG B.4: Wallaby

Quelle: <https://github.com/keathley/wallaby>

B.9 Risiken

Die Risikobewertung erfolgt mit folgender Formel:

$$\text{Bewertung} = \text{Schaden} \times \text{Eintrittswahrscheinlichkeit}$$

Schadensskala:

Gewichtung	Beschreibung
Gering (1-2)	Kleiner Schaden, hat kaum Auswirkungen auf das Projekt.
Mittel (3-4)	Mittlerer Schaden, Zeitverzögerungen oder Qualitätsverluste.
Hoch (5-6)	Hoher Schaden, wichtige Arbeiten oder Phasen können nicht abgeschlossen werden, schlimmstenfalls ein Abbruch des Projekts.

TABELLE B.6: Risiken - Schadensskala

Eintrittswahrscheinlichkeitsskala:

Gewichtung	Beschreibung
Gering (1-2)	Kleine Eintrittswahrscheinlichkeit.
Mittel (3-4)	Mittlere Eintrittswahrscheinlichkeit.
Hoch (5-6)	Hohe Eintrittswahrscheinlichkeit.

TABELLE B.7: Risiken - Eintrittswahrscheinlichkeit

Handlungen um Risikobewertungen zu senken:

Handlung	Beschreibung
Akzeptanz	Das Eintreten eines Risiko wird wissentlich angenommen.
Transfer	Die Verantwortung von Risiken können an Dritte abgegeben werden.
Verminderung	Der Schaden oder die Eintrittswahrscheinlichkeit kann begrenzt oder reduziert werden.
Vermeidung	Es kann jeglichen Schaden vermieden werden.

TABELLE B.8: Risiken - Handlungen zur Senkung der Bewertung

B.9.1 Projektrisiken

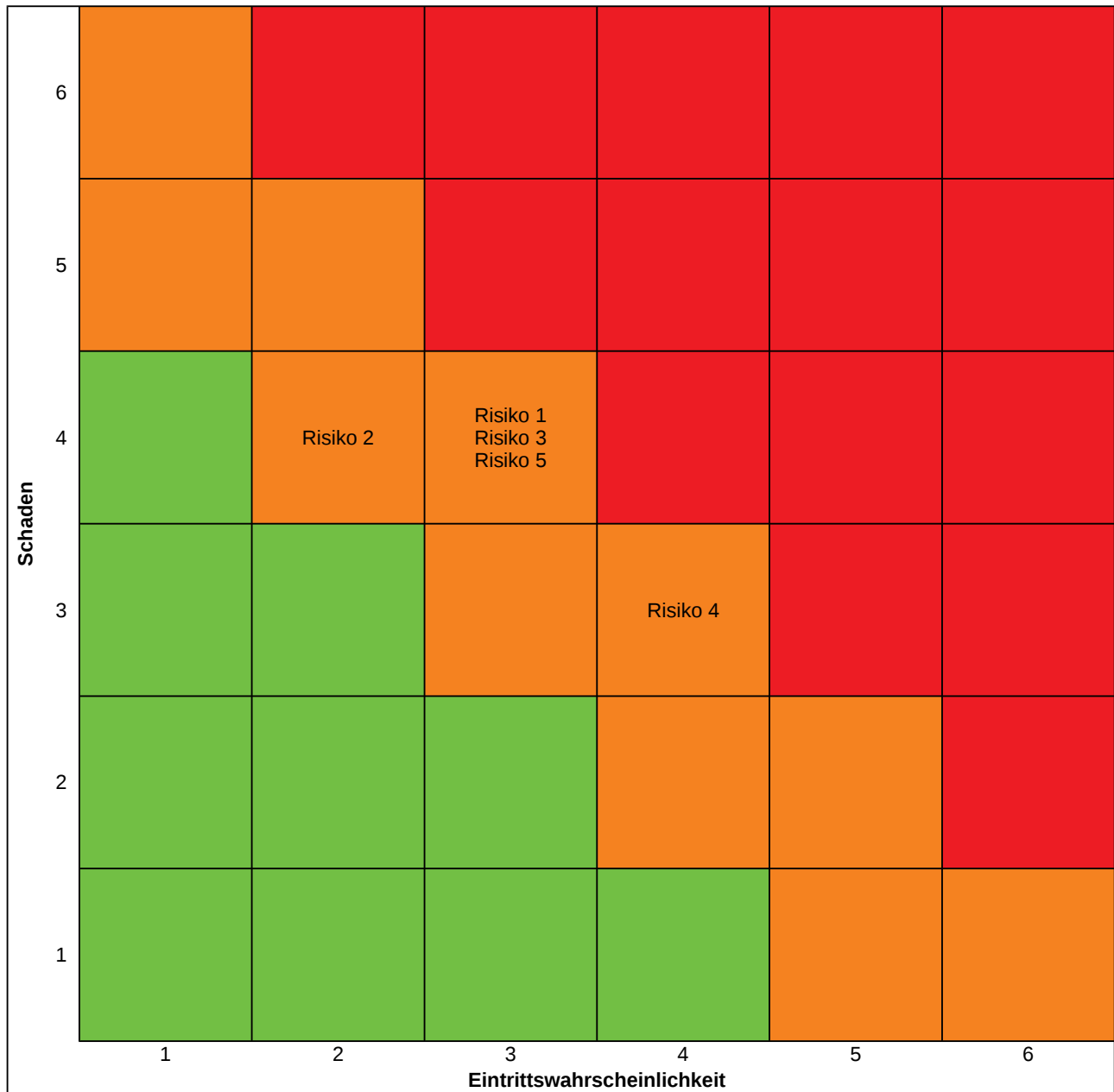
Nr.	Risiko	Auswirkung	Schaden	Wahrsch.	Bewertung
1	Ausfall des Entwicklers oder Projektleiters	Verzögerungen von Arbeiten	4	3	Mittel
2	Unvollständige Projektdokumentation	Schlechtere Diplomarbeit Bewertung	4	2	Mittel
3	Schlechter Projektplan	Verzögerungen und eventuelle Qualitätsverluste	4	3	Mittel
4	Keine Benutzer	Das Produkt wird nicht von Benutzern eingesetzt	3	4	Mittel
5	Technisch nicht umsetzbare Features	Das Produkt kann nicht wie angedacht benutzt werden	4	3	Mittel

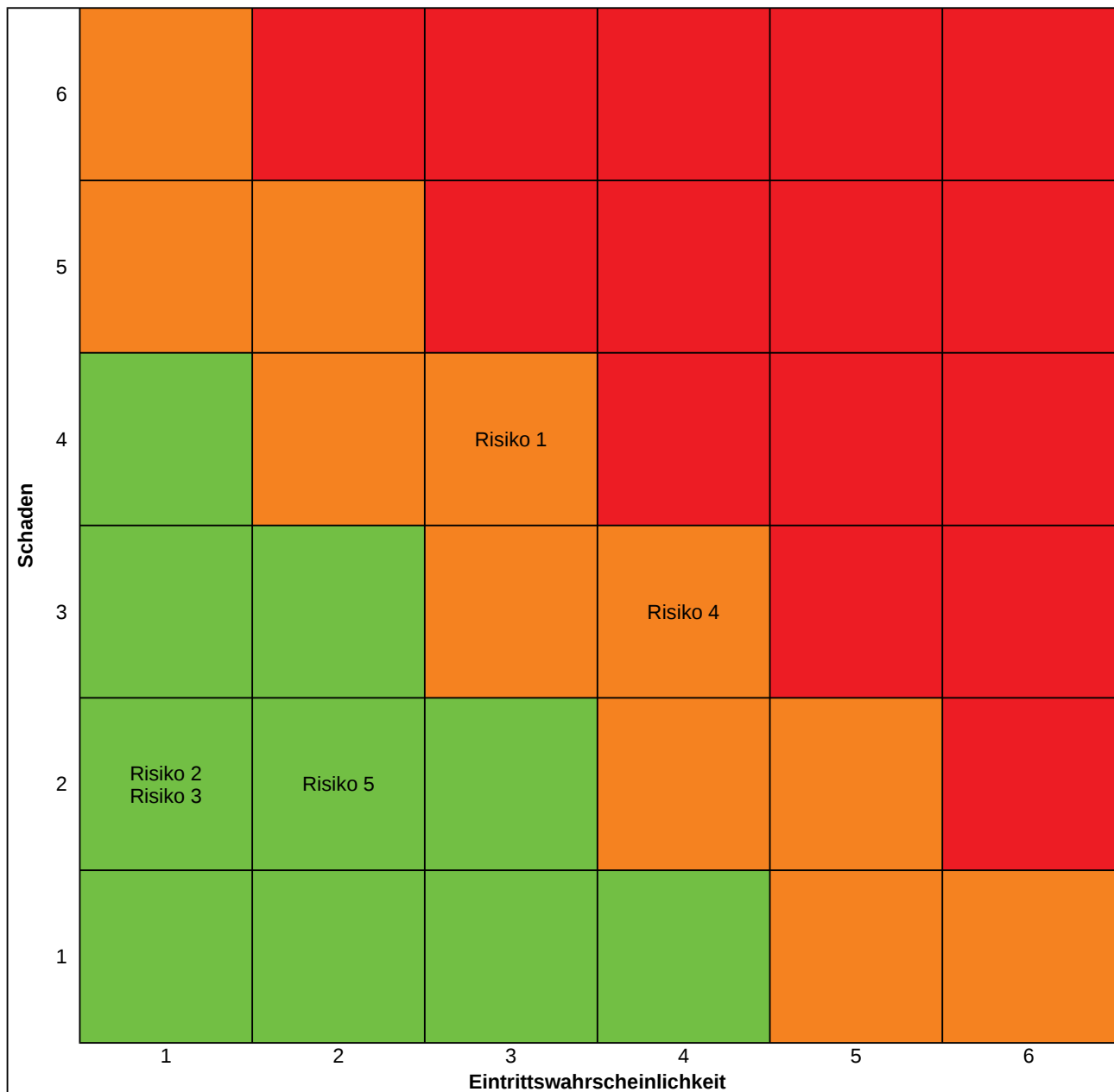
TABELLE B.9: Projektrisiken

B.9.2 Massnahmen

Nr.	Massnahme	Handlung	Bewertung nach Massnahme		
			Schaden	Wahrsch.	Bewertung
1	Arzt aufsuchen, ggf. Projekt-Pause oder Abbruch	Akzeptanz	4	3	Mittel
2	Statusbericht alle zwei Wochen, bei Fragen sofort Hilfe suchen	Verminderung	2	1	Gering
3	Genügend Buffer-Zeit einplanen, ggf. Ferientage für Projekt einsetzen	Verminderung	2	1	Gering
4	Das Produkt löst vor allem ein persönliches Interesse	Akzeptanz	3	4	Mittel
5	Vereinfachte Alternativen in Konzept-Phase untersuchen	Verminderung	2	2	Gering

TABELLE B.10: Projektrisiken

B.9.3 Risikodiagramm ohne Massnahmen

B.9.4 Risikodiagramm mit Massnahmen

Anhang C

Terminplan

Projektplan: Konzertkalender

Aktivität	Dauer [h]			Status	Wer																												
						Februar			März			April				Mai				Juni													
	Soll	Ist	Abw.			06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27						
Initialisierung	60	0	-60																														
1.1 Projektinitialisierung erstellen	4		-4	erledigt	DS																												
1.2 Projektorganisation	2		-2	erledigt	DS																												
1.3 Projektziele und Abgrenzungen	4		-4	erledigt	DS																												
1.4 Vorbereitung Kick Off & Meeting	8		-8	erledigt	DS,SB,SR																												
1.5 Projektplan	12		-12	erledigt	DS																												
1.6 Anforderungskatalog	4		-4	erledigt	DS																												
1.7 Risikoanalyse	4		-4	geplant	DS																												
1.8 Varianten beschreiben	8		-8	geplant	DS																												
1.9 Varianten evaluieren & auswählen	2		-2	geplant	DS																												
1.10 Projektauftrag erstellen	12		-12	geplant	DS																												
Konzept	66	0	-66																														
3.1 Portalnamen finden	2		-2	geplant	DS																												
3.2 Screens definieren	8		-8	geplant	DS																												
3.3 Screens designen	24		-24	geplant	DS,JS																												
3.4 Software Architektur	12		-12	geplant	DS																												
3.5 Test Konzept	12		-12	geplant	DS																												
3.6 Zwischen-Meeting	8		-8	geplant	DS,SB,SR																												
Realisierung	136	0	-136																														
4.1 Screens in HTML/CSS umsetzen	24		-24	geplant	DS																												
4.2 Initialisierung Backend	8		-8	geplant	DS																												
4.3 Implementation Registrierung/Login	8		-8	geplant	DS																												
4.4 Implementation Passwort Reset	8		-8	geplant	DS																												
4.5 Implementation der Screens	24		-24	geplant	DS																												
4.6 Implementation Suche	16		-16	geplant	DS																												
4.7 Tests erstellen	48		-48	geplant	DS																												
Abschluss	36	0	-36																														
5.1 Management Summary	4		-4	geplant	DS																												
5.2 Bericht ausdrucken, binden & senden	8		-8	geplant	DS																												
5.3 Diplomarbeit bewerten	16		-16	geplant	SB,SR																												
5.4 Abschluss Meeting	8		-8	geplant	DS,SB,SR																												
Total / bereits benötigt / Restliche Stunden:	286	0	-286																														

	286	0	286
Total Soll:			
Bereits benötigt:			
Restliche Stunden:			

Legende

Name	Abk.
Damian Senn	DS
Sandro Bertolino	SB
Severin Rätz	SR
Joshua Schär	JS

planung-empty.ods

Damian Senn

Anhang D

Studie

D.1 Zweck des Dokuments

In der Studie werden die Anforderungen aufgenommen, sowie Variantenbeschriebe für die Projektrealisierung erstellt. Die Varianten werden miteinander verglichen und durch den Variantenentscheid wird das weitere Vorgehen definiert. Ausserdem werden in der Studie die Risiken und Wirtschaftlichkeit des Projekts analysiert.

Folgende Arbeiten werden in dieser Studie abgehandelt:

- der Anforderungskatalog wird definiert
- die Evaluation der Browser Software-Technologien
- die Evaluation der Server Software-Technologien
- die Evaluation der Testing Software-Technologien
- eine Kostenschätzung und mögliche Wirtschaftlichkeit ausgerechnet

D.2 Informationsbeschaffung

Quelle	Beschreibung
Schulwissen / Berufserfahrung	Die Grundlage für die Umsetzung dieses Projekts wird durch mein existierendes Schulwissen sowie meine langjährige Berufserfahrung in der Software-Entwicklung gesetzt.
Internet	Ein Grossteil der Informationen werden heute über das Internet bezogen, für die Evaluation von Technologien und Lösungsansätzen wird einiges über das Internet recherchiert werden müssen.

TABELLE D.1: Informationsbeschaffung

D.3 Anforderungskatalog

Feature	Titel	Nr.	Kriterium	Ziel	Muss
Suche	Suche nach Konzertname	1.1	Listet alle Konzerte die Wörter der Suche im Konzertnamen beinhalten	1.1	Muss
	Suche nach Konzertlocation	1.2	Schränkt die Such-Resultate nach gegebener Konzertlocation ein	1.2	Muss
	Suche nach Ort	1.2	Schränkt die Such-Resultate nach gegebenem Ort ein	1.2	Muss
	Suche nach Genre	1.2	Schränkt die Such-Resultate nach gegebenem Musik-Genre ein	1.2	Muss
Design	Desktop	2.1	Alle Ansichten haben eine Desktop-Optimierte Variante	1.4	Muss
	Tablet	2.2	Alle Ansichten haben eine Tablet-Optimierte Variante	1.4	Muss
	Mobile	2.3	Alle Ansichten haben eine Mobile-Optimierte Variante	1.4	Muss
	Browser Kompatibilität	2.4	Alle Ansichten müssen in aktuellem Google Chrome und Mozilla Firefox dem Grundlayout folgen	1.9	Muss
SEO	Indexierbarkeit	3.1	Das Produkt ist von Suchmaschinen indexierbar	1.5	Muss
	Linked Data	3.2	Konzert Detailseiten sind mit dem Event-Schema ¹ ausgestattet	1.5	Muss
Benutzer	Registrierung	4.1	Besucher können sich einen Benutzer registrieren, Benutzernamen und E-Mail Adressen müssen einzigartig sein	1.6	Muss
	Passwort-Vergessen	4.2	Benutzer können sich einen Passwort-Reset Link anfordern	1.7	Muss
	Social	4.3	Benutzer können auf Konzerten vermerken ob sie Teilnehmen oder nicht	1.11	Kann

¹<https://schema.org/Event>

Feature	Titel	Nr.	Kriterium	Ziel	Muss
Erfassung	Artist	5.1	Benutzer können Artisten mit einem Genre erfassen	1.8	Muss
	Location	5.2	Benutzer können eine Konzertlocation mit Ort/Strasse erfassen	1.8	Muss
	Konzert	5.3	Benutzer können ein Konzert mit Konzertlocation und Artisten erfassen	1.8	Muss
	Facebook	5.4	Benutzer können ein Konzert in ein Facebook-Event exportieren	1.10	Kann
Security	SQL-Injection	6.1	Das Produkt soll resistent gegen SQL-Injection sein	1.12	Muss
	HTML-Injection	6.2	Das Produkt soll resistent gegen HTML-Injection / XSS sein	1.12	Muss
	Passwort encryption	6.3	Passwörter von Benutzer müssen mit einem sicheren Verfahren gespeichert werden	1.12	Muss
	Session	6.4	Session-Cookies dürfen nicht durch JavaScript ausgelesen werden	1.12	Kann
Performance	Ladezeit	7.1	Die Seitenansichten dürfen nicht länger als 6 Sekunden auf einem 3G Netz laden		Muss

TABELLE D.2: Anforderungskatalog

D.4 Evaluation Browser-Technologie

Kriterium	Gewicht	Abnahmekriterium
Komplexität	3	Die Technologie sollte im Rahmen der Diplomarbeit nicht eine zu hohe Komplexität vorweisen. Durch eine niedrigere Komplexität bestehen weniger Risiken dass technische Probleme auftreten werden.
Performance	4	In den Projektzielen wurde definiert, dass die Applikation in maximal 6 Sekunden im Browser geladen sein muss. Daher ist es wichtig, dass die Technologie gute Performance Charakteristiken vorweist.
SEO	5	Für eine öffentliche Applikation ist es unentbehrlich, dass sie indexierbar durch Suchmaschinen ist.
Interaktivität	4	Applikationen im Browser werden immer interaktiver, daher ist es wichtig, dass die Technologie anspruchsvolle Abläufe implementieren kann.
Stabilität	3	Für das Projekt ist es wichtig, dass auf eine stabile Technologie gesetzt wird, welche den Projektablauf so wenig wie möglich beeinträchtigt.
Testing	3	Durch einfaches Testing, kann sichergestellt werden, dass die Applikation wie gewünscht umgesetzt wurde und auch beim Weiterentwickeln nicht existierende Funktionalitäten beeinträchtigt werden.

TABELLE D.3: Browser-Technologie Kriterien

D.4.1 Variante: React

Die JavaScript Library **React** ist heute die wohl beliebteste Technologie um interaktive Applikationen im Web zu bauen.

D.4.2 Variante: Next.js

Next.js ist ein JavaScript Framework, das auf der **React** Library aufbaut und zusätzliche Features sowie gängige Konventionen mitbringt.

D.4.3 Variante: SSR

SSR steht für **Serverside Rendering** und beschreibt die klassische Methode vom Erstellen von Webseiten, indem man HTML auf dem Server generiert und zum Browser schickt.

Dies hat nach wie vor seine Daseinsberechtigung, da dies weniger Komplexität mit sich bringt, einen schnelleren Seitenaufbau garantiert und ohne zusätzlichen Aufwand von Suchmaschinen indexiert wird.

D.5 Bewertungen Browser-Technologie

Bewertung:

4 = Sehr gut, 3 = Gut, 2 = Ungenügend, 1 = Schlecht

Gewichtung:

5 = Unverzichtbar, 4 = Sehr wichtig, 3 = Erleichtert die Arbeit, 2 = Weniger wichtig, 1 = unwichtig

$$\text{Bewertung} \times \text{Gewichtung} = \text{Punktzahl}$$

Kriterium	Gewichtung	Variante: React		Variante: Next.js		Variante: SSR	
		Bewertung	Punkte	Bewertung	Punkte	Bewertung	Punkte
Komplexität	3	2	6	3	9	4	12
Performance	4	3	12	3	12	4	16
SEO	5	2	10	4	20	4	20
Interaktivität	4	4	16	4	16	3	12
Stabilität	3	2	6	3	9	4	12
Testing	4	4	16	4	16	4	16
Total:		React:	66	Next.js:	82	SSR:	88

TABELLE D.4: Browser-Technologie Bewertung

D.6 Entscheid Browser-Technologie

Durch die Evaluierung wurde klar, dass das Einsetzen eines JavaScript-Frameworks zuviel zusätzliche Komplexität und gewisse einbussungen in Performance und Stabilität unvermeidbar ist. Somit ist ein die Wahl für eine klassische Server-Side Rendered Webseite favorisierend.

Es ist durchaus vorstellbar, dass in einem zweiten Schritt, nach diesem Projekt, die Server-Side Rendered Applikation durch eine Next.js Applikation ersetzt werden könnte.

D.7 Evaluation Server-Technologie

Kriterium	Gewicht	Abnahmekriterium
Komplexität	3	Die Technologie sollte im Rahmen der Diplomarbeit nicht eine zu hohe Komplexität vorweisen. Durch eine niedrigere Komplexität bestehen weniger Risiken dass technische Probleme auftreten werden.
Performance	4	In den Projektzielen wurde definiert, dass die Applikation in maximal 6 Sekunden im Browser geladen sein muss. Daher ist es wichtig, dass die Technologie gute Performance Charakteristiken vorweist.
Stabilität	5	Während es für die Browser-Technologie vorstellbar ist, die Technologie auszuwechseln, ist es für den Server wichtig auf eine stabile und zukunftssichere Technologie zu setzen.
Testing	5	Durch einfaches Testing, kann sichergestellt werden, dass die Applikation wie gewünscht umgesetzt wurde und auch beim Weiterentwickeln nicht existierende Funktionalitäten beeinträchtigt werden. Vorallem auf dem Server ist wichtig, dass die Businesslogik gut abdeckend getestet werden kann.

TABELLE D.5: Server-Technologie Kriterien

D.7.1 Variante: Node.js / koa.js

Auch auf dem Server gewinnt JavaScript immer mehr an Beliebtheit. Mit Node.js und koa.js können schnell kleinere und simplere Applikationen erstellt werden, die dennoch sehr performant sind.

D.7.2 Variante: Elixir / Phoenix

Elixir ist eine Programmiersprache die eine sehr stabile und performante Grundlage bietet. Durch das Framework Phoenix, wird im Elixir Ökosystem ein starkes feature umfangreiches Web-Framework angeboten.

D.7.3 Variante: Next.js

Next.js wurde bereits als Variante für die Browser-Technologie in Betracht gezogen. Ein zusätzliches Feature von Next.js ist, dass die Applikation auch auf dem Server betrieben werden kann. Das Einsetzen der selben Technologie kann bedeutende Vorteile mit sich bringen, so muss man nur ein Framework lernen und kann Programmcode auf dem Server mit der Applikation im Browser geteilt werden.

D.8 Bewertungen Server-Technologie

Bewertung:

4 = Sehr gut, 3 = Gut, 2 = Ungenügend, 1 = Schlecht

Gewichtung:

5 = Unverzichtbar, 4 = Sehr wichtig, 3 = Erleichtert die Arbeit, 2 = Weniger wichtig, 1 = unwichtig

$$\text{Bewertung} \times \text{Gewichtung} = \text{Punktzahl}$$

Kriterium	Gewichtung	Variante: koa.js		Variante: Phoenix		Variante: Next.js	
		Bewertung	Punkte	Bewertung	Punkte	Bewertung	Punkte
Komplexität	3	2	6	4	12	3	9
Performance	4	4	16	4	16	3	12
Stabilität	5	3	15	4	20	3	15
Testing	5	3	15	4	20	4	20
Total:		koa.js:	55	Phoenix:	68	Next.js:	56

TABELLE D.6: Server-Technologie Bewertung

D.9 Entscheid Server-Technologie

Durch das grosse Featureset von Phoenix sowie tiefer Komplexität gegenüber den beiden anderen Varianten hat sich Phoenix für die Server-Technologie ganz klar durchgesetzt.

D.10 Evaluation Testing-Technologie

Kriterium	Gewicht	Abnahmekriterium
Performance	3	Bei wachsender Anzahl von Tests ist es wichtig, dass die Test-Software genug skalierbar ist um Tests in parallel auszuführen.
Stabilität	5	
Backend-Integration	4	Es ist sehr hilfreich, wenn die End-to-End Test-Software vom Server direkt ausgeführt werden. So kann gleichzeitig zum Browser-Test auch die Businesslogik getestet werden.
Visualtesting	5	Die Technologie soll mit dem Service percy.io integrierbar sein.

TABELLE D.7: Testing-Technologie Kriterien

D.10.1 Jest + Puppeteer

Jest ist ein JavaScript-Test Framework von Facebook. Durch die Kombination der Puppeteer Library von Google ist es möglich, automatisierte Browser-Tests durchzuführen.

D.10.2 Wallaby

Wallaby ist ein Elixir Browser-Test Framework, welches sich nahtlos mit Phoenix integrieren lässt. Wallaby unterstützt parallelisierung von Tests und ist daher ein guter Kandidat eine hohe Anzahl von automatisierten Tests.

D.11 Bewertungen Testing-Technologie

Bewertung:

4 = Sehr gut, 3 = Gut, 2 = Ungenügend, 1 = Schlecht

Gewichtung:

5 = Unverzichtbar, 4 = Sehr wichtig, 3 = Erleichtert die Arbeit, 2 = Weniger wichtig, 1 = unwichtig

Bewertung x Gewichtung = Punktzahl

Kriterium	Gewichtung	Variante: Jest		Variante: Wallaby	
		Bewertung	Punkte	Bewertung	Punkte
Performance	3	4	12	4	12
Stabilität	5	3	15	4	20
Backend-Integration	4	2	8	4	16
Visualtesting	4	4	16	1	4
Total:		Jest:	51	Wallaby:	52

TABELLE D.8: Testing-Technologie Bewertung

D.12 Entscheid Testing-Technologie

Dadurch dass sich Wallaby einfach mit der ausgewählten Server-Technologie verwenden lässt, hohe Performance und Stabilität aufweist, ist Wallaby die knapp bessere Variante als eine Jest + Puppeteer kombination.

Leider hat Wallaby keine Visualtesting Integration mit dem Dienst percy.io, dies könnte aber im verlaufe der Umsetzung eventuell im Rahmen dieser Arbeit umgesetzt werden.

D.13 Wirtschaftlichkeit

D.13.1 Break Even Analyse

Anhang E

Konzept

E.1 Design- und Bedienkonzept

E.2 Softwarekonzept

E.3 Testkonzept

Anhang F

Arbeitsjournal

F.1 Sonntag 3. März

2h:

- Vorbereitung Kick-off
- Abgrenzung erweitern
- Grobe Anforderungen
- Auflistung möglicher Variantenentscheide
- TODO ergänzt

F.2 Dienstag 5. März

2h:

- Vorbereitung Kick-off

F.3 Mittwoch 6. März

3h:

- Vorbereitung Sitzungszimmer
- Kick-off Meeting

4h:

- An Projektauftrag arbeiten - Auftraggeber geändert nach Empfehlung von Marc Aeby

F.4 Samstag 9. März

2h:

- An Studie/Pflichtenheft arbeiten

F.5 Dienstag 12. März

2h:

- PDF Generierung und Ordnerstruktur angepasst

F.6 Samstag 16. März

3h:

- Projektplan von gantt nach ods migrieren

F.7 Dienstag 19. März

0.5h:

- Projektplan in Berichtanhang angehängt

F.8 Mittwoch 27. März

1.5h:

- Projektplan an korrekte HERMES 5 Struktur angepasst
- Titelblatt von Ilias hinzugefügt
- Termin am 12.04.2019 mit Joshua Schär für einen Screendesign-Workshop abgemacht

F.9 Sonntag 31. März

5h:

- Projektziele erweitert
- Anforderungskatalog erweitert

F.10 Sonntag 31. März

8h:

- Definitive Projektziele definiert
- Anforderungskatalog fertig gestellt
- Gesamte Berichtstruktur ausgelegt, Ziele und Abgrenzungen in Bericht hinterlegt und referenziert

F.11 Freitag 5. April

2h:

- An Studie weiter gearbeitet
- Variantenkriterien definiert

F.12 Samstag 6. April

4h:

- An Studie weiter gearbeitet
- Variantenbeschreibungen erstellt
- Variantenbewertungen

F.13 Mittwoch 10. April

2h:

- Brainstorming für Portalnamen