

Konzertkalender

Damian Senn

TSBE

DIPLOMARBEIT

Konzertkalender

Author:
Damian Senn

Experten:
Sandro Bertolino
Severin Rätz

Eine Diplomarbeit für den Abschluss Dipl. Techniker Informatik

17. 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 Mögliche Varianten	9
1.9.4 Evaluation Varianten	9
1.9.5 Entscheid Varianten	9
1.9.6 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 Sitzungsprotokoll - Kickoff	18
C Terminplan	19
D Studie	20
D.1 Zweck des Dokuments	20
D.2 Informationsbeschaffung	20
D.3 Anforderungskatalog	21
D.4 Evaluation Browser-Technologie	23
D.4.1 Variante: React	24
D.4.2 Variante: Next.js	24
D.4.3 Variante: SSR	24
D.5 Bewertungen Browser-Technologie	25
D.6 Entscheid Browser-Technologie	25
D.7 Evaluation Server-Technologie	26
D.7.1 Variante: Node.js / koa.js	26
D.7.2 Variante: Elixir / Phoenix	26
D.7.3 Variante: Next.js	27
D.8 Bewertungen Server-Technologie	28
D.9 Entscheid Server-Technologie	28
D.10 Evaluation Testing-Technologie	29
D.10.1 Jest + Puppeteer	29
D.10.2 Wallaby	29
D.11 Bewertungen Testing-Technologie	30
D.12 Entscheid Testing-Technologie	30
D.13 Wirtschaftlichkeit	31
D.13.1 Projektkosten	31
D.13.2 Break Even Analyse	32
E Projektauftrag	34
E.1 Zweck des Dokuments	34
E.2 Ausgangslage	34
E.3 Projektziele	35
E.4 Rahmenbedingungen	35
E.5 Terminplan	36
E.6 Meilensteine	36
E.7 Organigramm	37
E.7.1 Tätigkeiten im Projekt	37
E.7.2 Kommunikation	37
E.8 Abgrenzungen	38
E.9 Anforderungskatalog	39
E.10 Lösungsbeschreibung	41
E.11 Kosten	42
E.12 Risiken	43
E.12.1 Projektrisiken	44
E.12.2 Massnahmen	45
E.12.3 Risikodiagramm ohne Massnahmen	46
E.12.4 Risikodiagramm mit Massnahmen	47
F Wirtschaftlichkeit - Gigboost	48

G Wirtschaftlichkeit - Werbung	49
H Konzept	50
H.1 Portalname	50
H.2 Design- und Bedienkonzept	51
H.2.1 Mockups	51
H.3 Softwarekonzept	56
H.3.1 Datenbankstruktur	56
H.4 Testkonzept	56
I Arbeitsjournal	57
I.1 Sonntag 3. März	57
I.2 Dienstag 5. März	57
I.3 Mittwoch 6. März	57
I.4 Samstag 9. März	57
I.5 Dienstag 12. März	57
I.6 Samstag 16. März	58
I.7 Dienstag 19. März	58
I.8 Mittwoch 27. März	58
I.9 Sonntag 31. März	58
I.10 Sonntag 31. März	58
I.11 Freitag 5. April	58
I.12 Samstag 6. April	58
I.13 Mittwoch 10. April	59
I.14 Freitag 12. April	59
I.15 Samstag 13. April	59
I.16 Sonntag 14. April	59
I.17 Montag 15. April	59
J Biweekly Reports	60
J.1 Kalenderwoche 11-12	60
J.2 Kalenderwoche 13-14	61
J.3 Kalenderwoche 15-16	62

Abbildungsverzeichnis

1.1	Organigram	3
1.2	Abgrenzungen	6
D.1	Break-Even Analyse - Gigboost	32
D.2	Break-Even Analyse - Werbung	33
E.1	Organigram	37
E.2	Abgrenzungen	38
E.3	Phoenix Framework Logo	41
E.4	Wallaby Logo	41
H.1	Mockup: Homepage	51
H.2	Mockup: Suchresultate	52
H.3	Mockup: Gig Ansicht	53
H.4	Mockup: Gig erfassen	54
H.5	Mockup: Benutzerprofil	55
H.6	Entity Relationship Diagram	56

Tabellenverzeichnis

1.1	Ziele	2
1.2	Informationsbeschaffung	7
1.3	Anforderungskatalog	9
D.1	Informationsbeschaffung	20
D.2	Anforderungskatalog	22
D.3	Browser-Technologie Kriterien	23
D.4	Browser-Technologie Bewertung	25
D.5	Server-Technologie Kriterien	26
D.6	Server-Technologie Bewertung	28
D.7	Testing-Technologie Kriterien	29
D.8	Testing-Technologie Bewertung	30
D.9	Projektkosten	31
D.10	Betriebskosten	31
D.11	Werbeeinnahmen pro Besucher	33
E.1	Ziele	35
E.2	Terminplan	36
E.3	Meilensteine	36
E.4	Tätigkeiten Verteilung	37
E.5	Anforderungskatalog	40
E.6	Projektkosten	42
E.7	Betriebskosten	42
E.8	Risiken - Schadensskala	43
E.9	Risiken - Eintrittswahrscheinlichkeit	43
E.10	Risiken - Handlungen zur Senkung der Bewertung	43
E.11	Projektrisiken	44
E.12	Projektrisiken - Massnahmen	45

Abkürzungsverzeichnis

TSBE	T elematik S chule B ern
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
CPC	C ost P er C lick
CPM	C ost P er M ile

For/Dedicated to/To my...

Kapitel 1

Initialisierung

1.1 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. Den Benutzern der Plattform soll es möglich sein, Konzerte selber zu erfassen und pflegen.

Um einen zusätzlichen Service für den Benutzer 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.

1.2 Projektziele

Folgende Ziele sind in der Initialisierungsphase definiert worden:

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 1 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 E.7.2 beschrieben.	Muss
2.5	Das Projekt muss bis Ende Mai 2019 abgeschlossen sein	Muss

TABELLE 1.1: Ziele

1.3 Projektorganisation

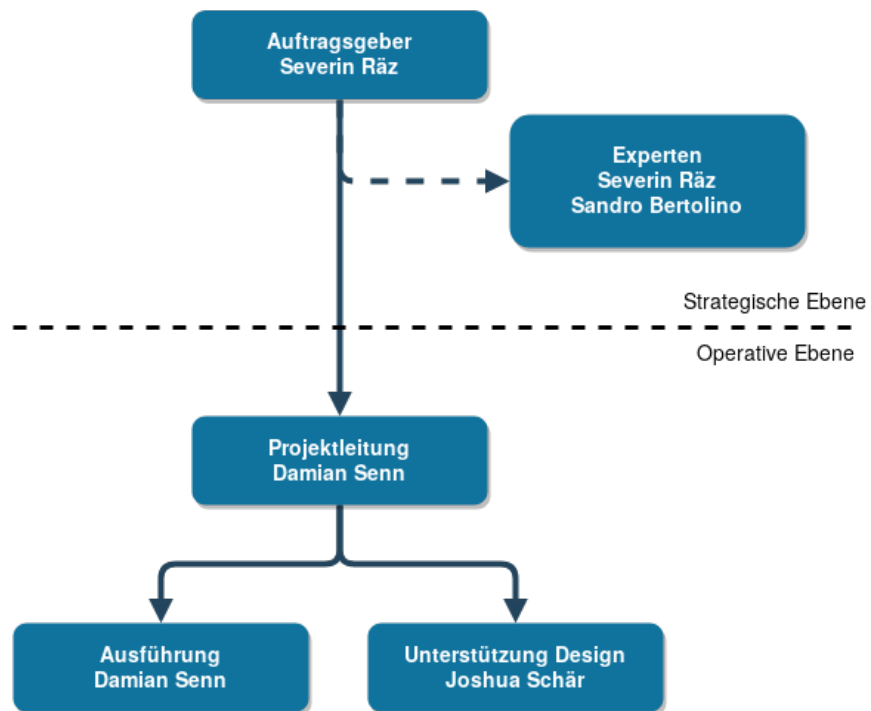


ABBILDUNG 1.1: Organigramm

1.4 Ausgefüllter Projektplan

Projektplan: Konzertkalender

Aktivität	Dauer [h]			Status	Wer																								
	Soll	Ist	Abw.			Februar				März				April				Mai				Juni							
						06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27		
Initialisierung	64	71	7																										
1.1 Projektinitialisierung erstellen	4	4	0	erledigt	DS																								
1.2 Projektorganisation	2	2	0	erledigt	DS																								
1.3 Projektziele und Abgrenzungen	4	4	0	erledigt	DS																								
1.4 Vorbereitung Kick Off & Meeting	8	8	0	erledigt	DS,SB,SR																								
1.5 Projektplan	12	16	4	erledigt	DS																								
1.6 Anforderungskatalog	4	6	2	erledigt	DS																								
1.7 Risikoanalyse	4	4	0	erledigt	DS																								
1.8 Varianten beschreiben	8	7	-1	erledigt	DS																								
1.9 Varianten evaluieren & auswählen	2	2	0	erledigt	DS																								
1.10 Wirtschaftlichkeit evaluieren	4	6	2	erledigt	DS																								
1.11 Projektauftrag erstellen	12	12	0	erledigt	DS																								
Konzept	66	16	-50																										
3.1 Portalnamen finden	2	2	0	erledigt	DS																								
3.2 Screens definieren	8	4	-4	erledigt	DS																								
3.3 Screens designen	24	8	-16	in arbeit	DS,JS																								
3.4 Software Architektur	12	2	-10	in arbeit	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				69				-217															
						286				69				217															
						Total Soll:				Bereits benötigt				Restliche Stunden:															

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

1.5 Lieferergebnisse

- Studie (Anhang **D**)
- Projektauftrag (Anhang **E**)
- Konzept (Anhang **H**)

1.6 Ressourcenplan

1.7 Risiken

1.8 Abgrenzungen

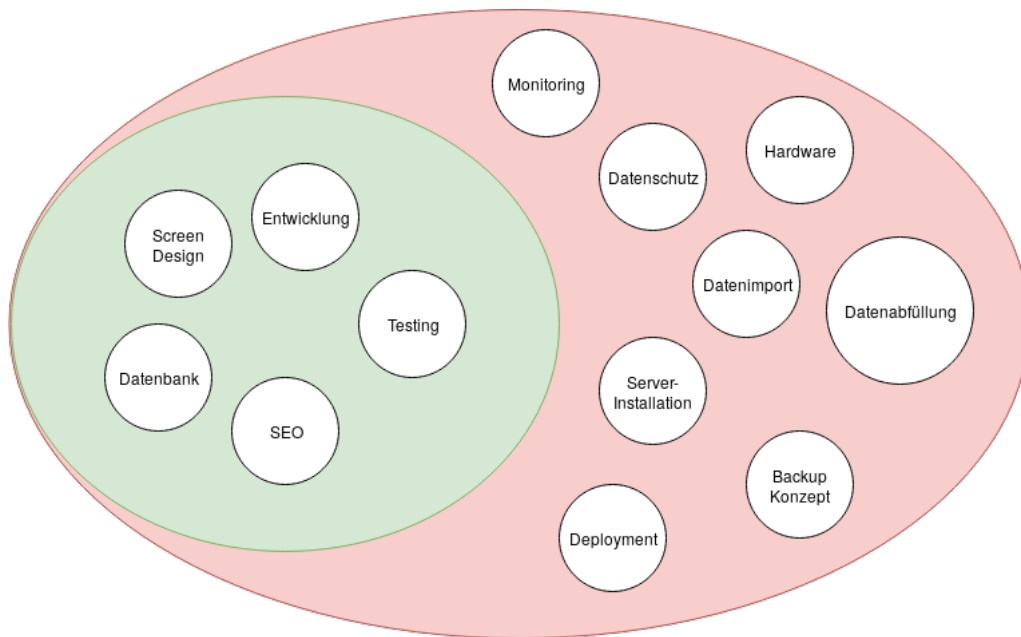


ABBILDUNG 1.2: Abgrenzungen

Die detaillierten Erklärung zu den Abgrenzungen sind im Projektauftrag [E.8](#) zu finden.

1.9 Studie

1.9.1 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.
Externer Experte	Bei konzeptionellen sowie technischen Fragen kann der externe Experte um Rat gefragt werden.

TABELLE 1.2: 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/MusicEvent>

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
Sonstiges	User Tracking	8.1	Benutzerverhalten soll analysiert und nachvollziehbar sein.		Kann

TABELLE 1.3: Anforderungskatalog

1.9.3 Mögliche Varianten**1.9.4 Evaluation Varianten****1.9.5 Entscheid Varianten****1.9.6 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

PROJEKTINITIALISIERUNGSAUFTAG

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

Sitzungsprotokoll - Kickoff

Hallo Sandro, hallo Severin

Folgende Themen/ Aktionen haben wir am Kickoff Meeting besprochen:

- Der Diplombericht ist zwingend eine Woche vor dem Abschlussgespräch ausgedruckt abzugeben.
- Ich frage Marc Aeby betreffend der Phasenfreigaben, wie das funktioniert wenn ich Auftragsgeber sowie Projektleiter bin.
- Für die Wirtschaftlichkeit werde ich aufzeigen was das Projekt gekostet hat und zeige mit einer Break-Even Analyse auf, ab wann das Projekt potentiell Gewinn machen könnte.
- Ich werde mir noch einmal den Bewertungsbogen genau durchlesen
- Die Abgrenzungen werde ich noch einmal Anpassen und das Monitoring sowie Deployment aus dem Projekt abgrenzen.
- Die Meilensteine werden noch von mir definiert.
- Sandro hat in KW15 und KW16 viel los und ist anfangs Mai im Ausland.

Bei Fragen oder Anmerkungen stehe ich euch gerne zur Verfügung.

Gruss
Damian

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

Folgende Quellen werden in diesem Projekt für die Informationsbeschaffung genutzt:

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.
Externer Experte	Bei konzeptionellen sowie technischen Fragen kann der externe Experte um Rat gefragt werden.

TABELLE D.1: Informationsbeschaffung

D.3 Anforderungskatalog

Im Anforderungskatalog werden die Muss- und Kann-Kriterien definiert. Muss-Kriterien sind zwingend zu erfüllen, Kann-Kriterien sind als optionale Erweiterung zu verstehen.

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/MusicEvent>

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
Sonstiges	User Tracking	8.1	Benutzerverhalten soll analysiert und nachvollziehbar sein.		Kann

TABELLE D.2: Anforderungskatalog

D.4 Evaluation Browser-Technologie

Gewichtung:

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

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.

React ermöglicht es den Entwicklern Screens auf eine deklarative Weise zu definieren, so dass sich Elemente jeweils dem Zustand der Applikation anpassen.

Die Features von React sind minimal gehalten und sehen auf den ersten Blick einfach aus, jedoch wird für Anwendungen schnell klar, dass zusätzliche Software-Libraries benötigt werden um eine grössere Applikation zu entwickeln.

Durch das Hinzufügen von weiteren Libraries steigt die Komplexität dieser Variante stark an, da es im React Ökosystem für viele Lösungen diverse verschiedene Lösungsansätze gibt.

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.

Dadurch dass Next.js ein komplettes Framework ist und nicht nur eine Library, leidet diese Variante weniger der Komplexität eines kompletten Eigenbaus mit React. Features wie die Navigation von Seite zu Seite bringt Next.js bereits mit.

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 wenig Komplexität mit sich bringt, einen schnelleren Seitenaufbau garantiert und ohne zusätzlichen Aufwand von Suchmaschinen indexiert werden kann.

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.

Kosten / Wirtschaftlichkeit

Da die Programmiersprache Elixir sowie das Framework Phoenix unter einer Open Source Lizenz verfügbar sind, fallen bei dieser Lösung keine zusätzlichen Kosten an.

D.7 Evaluation Server-Technologie

Gewichtung:

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

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.

koa.js ist eine Library für Server-Applikationen und bietet nur eine einfache Basis und bringt keine Features für Datenpersistenz oder HTML Templates mit sich. Diese Features müssen durch zusätzliche Module installiert werden.

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. Phoenix beinhaltet eine grosse Menge an Features mit sich und gibt dem Entwickler direkt Funktionalitäten wie HTML Templates, Datenpersistenz und ein einfaches Test-Framework.

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 Featurset von Phoenix sowie tieferer Komplexität gegenüber den beiden anderen Varianten hat sich Phoenix für die Server-Technologie klar durchgesetzt.

Kosten / Wirtschaftlichkeit

Da die Programmiersprache Elixir sowie das Framework Phoenix unter einer Open Source Lizenz verfügbar sind, fallen bei dieser Lösung keine zusätzlichen Kosten an.

D.10 Evaluation Testing-Technologie

Gewichtung:

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

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

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

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.

Kosten / Wirtschaftlichkeit

Auch Wallaby ist unter eines Open Source Lizenz veröffentlicht und erzeugt somit im Projekt keine direkten zusätzlichen Kosten. Durch fehlende Unterstützung von Visualtesting, wird im Verlauf der Realisierung zusätzlicher Aufwand erzeugt, entweder durch eigene Implementation solcher Tests oder durch unbemerkte Fehler die sich bei Änderungen eingeschlichen haben.

D.13 Wirtschaftlichkeit

D.13.1 Projektkosten

Für die Berechnung der Projektkosten wird ein Stundensatz von 150.- CHF angenommen.

Phase	Geplante Stunden	Kosten
Initialisierung	64	9'600.- CHF
Konzept	66	9'900.- CHF
Realisierung	136	20'400.- CHF
Abschluss	64	5'400.- CHF
Total:	286	42'900.- CHF

TABELLE D.9: Projektkosten

Die geplanten Projektkosten betragen somit **42'900.- CHF**.

Kostenstelle	Jährliche Kosten
Software	Keine
.com Domain	20.- CHF
Hosting	1'800.- CHF
Total:	1'820.- CHF

TABELLE D.10: Betriebskosten

Für die Betriebskosten eines Hostings wird einen durchschnittlichen monatlichen Preis von 150.- CHF angenommen, da das Deployment für dieses nicht vorgesehen ist, ist dies eine von Damian Senn geschätzte Zahl.

D.13.2 Break Even Analyse

Gigboost

Beim Modell «Gigboost» wird Benutzern eine Option angeboten bei der ihre publizierten Gigs auf der Startseite sowie in Suchresultaten anderen Einträgen bevorzugt dargestellt werden. Für einen Gegenpreis von 10.- CHF kann ein Benutzer seinen Gig «boosten».

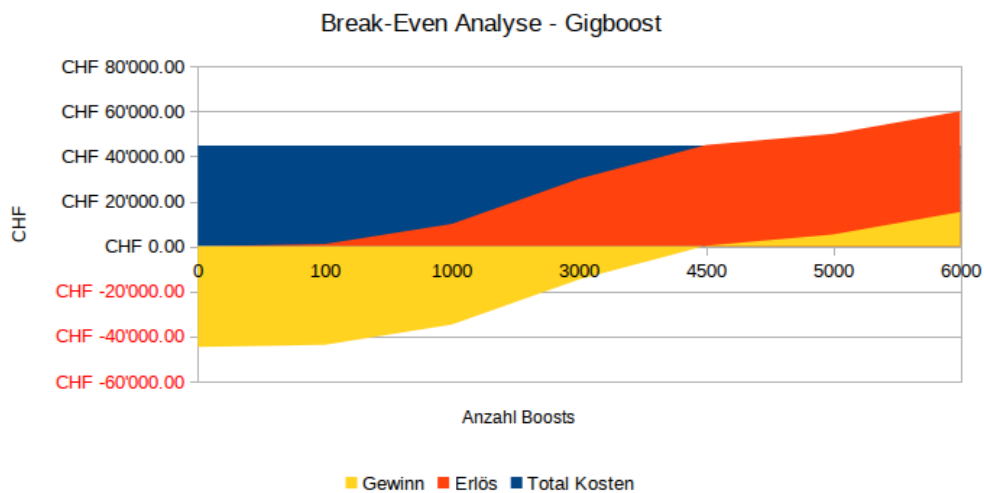


ABBILDUNG D.1: Break-Even Analyse - Gigboost

Werbung

Im Modell «Werbung» wird ausgerechnet wieviele aktive Benutzer das Produkt benötigen um in den nächsten Jahren Gewinn zu erzielen.

Durch Annahme von einem Erlös von 140.- CHF pro 40'000 Besucher² erhalten wir folgendes Bild:

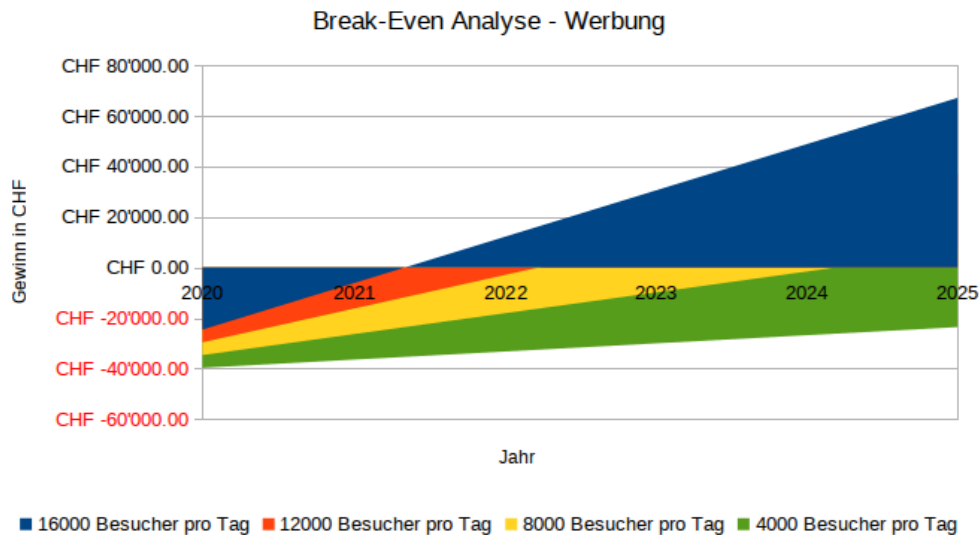


ABBILDUNG D.2: Break-Even Analyse - Werbung

Besucher pro Tag	Erlös pro Tag	Erlös pro Monat
2'000	7.- CHF	210.- CHF
4'000	14.- CHF	420.- CHF
8'000	28.- CHF	840.- CHF
12'000	42.- CHF	1'260.- CHF
16'000	46.- CHF	1'680.- CHF

TABELLE D.11: Werbeeinnahmen pro Besucher

Der Grafik ist zu entnehmen, dass das Produkt bei 8'000 Besucher pro Tag nach ca. 6 Jahren Gewinn erzielt. Bei 12'000 Besucher pro Tag erzielt das Produkt nach bereits 4 Jahren Gewinn und mit 16'000 Besucher pro Tag schon im dritten Jahr.

²<https://www.quora.com/How-much-does-Google-AdSense-pay-for-3-banners-on-a-webpage-per-1-000-views/answer/Manas-Sahu-59>

Anhang E

Projektauftrag

E.1 Zweck des Dokuments

Der Projektauftrag ist die verbindliche Vereinbarung zwischen Auftraggeber und Projektleiter, und bildet die Grundlage für den Projektstart sowie die Phasenfreigaben.

Folgend sind alle wichtigen Informationen die in der Phase Initialisierung erarbeitet wurden. Alle weiteren Details die in der Initialisierung ausgearbeitet wurden sind in der Studie im Anhang **D** zu finden.

E.2 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. Den Benutzern der Plattform soll es möglich sein, Konzerte selber zu erfassen und pflegen.

Um einen zusätzlichen Service für den Benutzer 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.

E.3 Projektziele

Folgende Ziele sind in der Initialisierungsphase definiert worden:

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 ¹ 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 E.7.2 beschrieben.	Muss
2.5	Das Projekt muss bis Ende Mai 2019 abgeschlossen sein	Muss

TABELLE E.1: Ziele

E.4 Rahmenbedingungen

- Das Projekt wird im Rahmen der Diplomarbeit durchgeführt.
- Die Richtlinien zum Erstellen des Diplomberichtes der TSBE. müssen eingehalten werden.
- Als Projektmethodik wird HERMES verwendet, angepasst auf das Projekt.
- Sämtliche Projekt-Dokumente sowie Programmcode wird regelmässig ins private Github Repository¹ geladen.

¹<https://github.com/topaxi/diplomarbeit-tsbe>

E.5 Terminplan

Nachfolgend ist der grobe Terminplan für die geplanten Phasen. Im Anhang E.5 ist der detaillierte Terminplan abgelegt.

Phase	Datum	Stunden
Initialisierung	06.03.2019 - 31.03.2019	64
Konzept	01.04.2019 - 21.04.2019	66
Realisierung	22.04.2019 - 19.05.2019	136
Abschluss	20.05.2019 - 26.05.2019	36
Total:		286

TABELLE E.2: Terminplan

E.6 Meilensteine

Im Projektplan wurden folgende Meilensteine und Termine festgelegt:

Nr.	Meilenstein	KW	Datum
1	Kickoff-Meeting	10	06.03.2019
2	Abschluss Phase Initialisierung	13	31.03.2019
3	Zwischen-Meeting	18	24.04.2019
4	Abschluss Phase Konzept	16	21.04.2019
5	Abschluss Phase Realisierung	20	19.05.2019
6	Abschluss Phase Abschluss	21	
7	Abschluss-Meeting	22	

TABELLE E.3: Meilensteine

Das Datum für das Abschluss-Meeting wird im Zwischen-Meeting mit den Experten, Sandro Bertolino und Severin Rätz, festgelegt. Der Abschluss der Phase Abschluss ist Abhängig vom Abschluss-Meeting und wird mindestens eine Woche vor dem Meeting stattfinden.

E.7 Organigramm

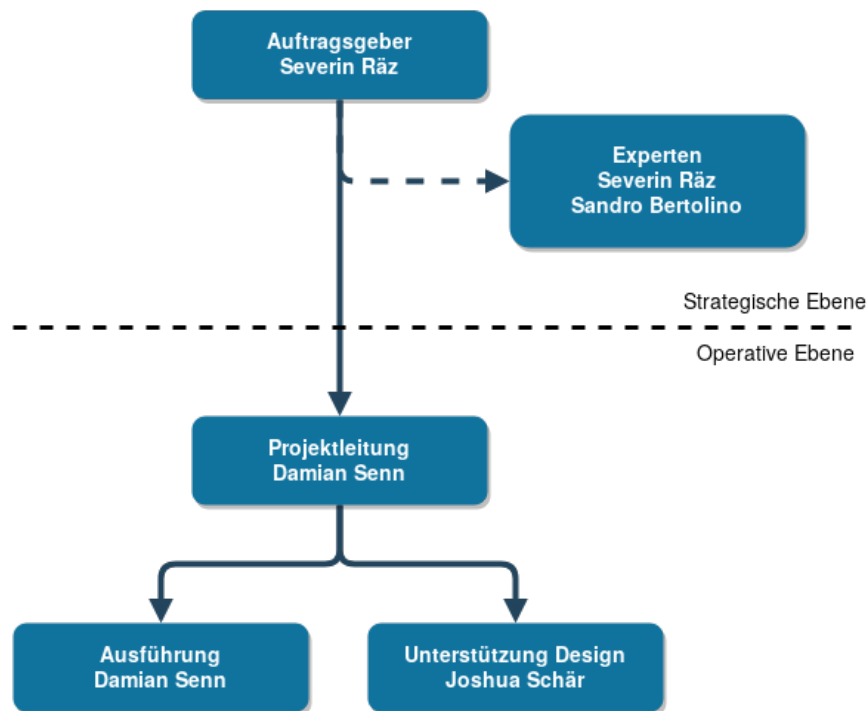


ABBILDUNG E.1: Organigramm

E.7.1 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
Joshua Schär	Unterstützung Design

TABELLE E.4: Tätigkeiten Verteilung

E.7.2 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.

E.8 Abgrenzungen

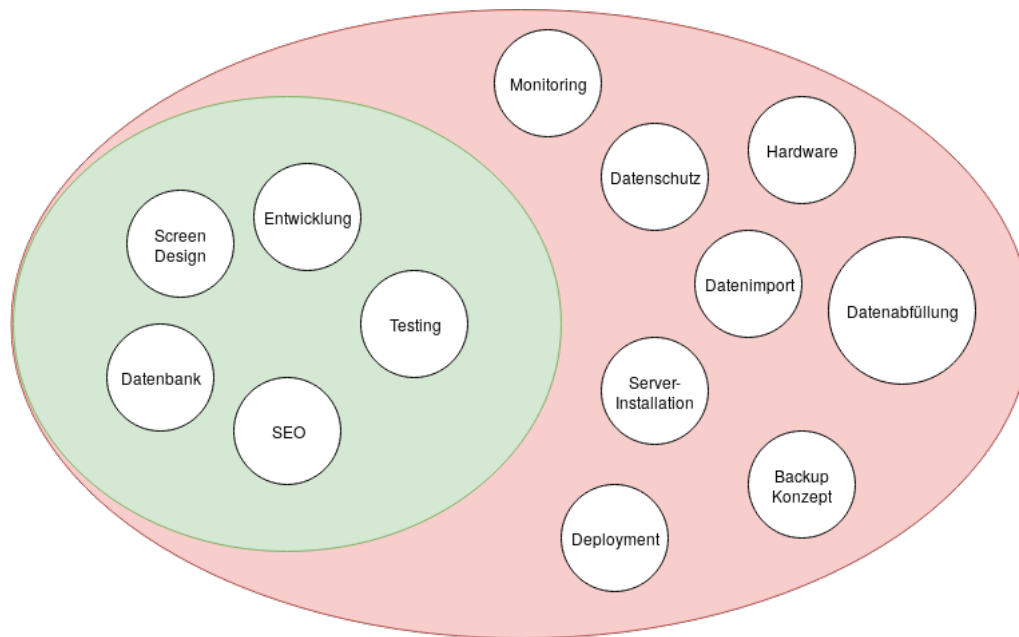


ABBILDUNG E.2: Abgrenzungen

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.

E.9 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 ² 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

²<https://schema.org/MusicEvent>

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
Sonstiges	User Tracking	8.1	Benutzerverhalten soll analysiert und nachvollziehbar sein.		Kann

TABELLE E.5: Anforderungskatalog

E.10 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 E.3: Phoenix Framework Logo

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

Die Nutzwertanalyse hat ergeben, dass es sinnvoller ist, das Projekt mit einer klassischen SSR Applikation zu starten. Das Phoenix Framework bietet alle benötigten Features an und kann durch Module einfach erweitert werden.

Für dynamische Interaktionen wie Formular-Validierungen wird zu einfachem JavaScript gegriffen. Ist ein Screen besonders interaktiv, kann gegebenenfalls eine kleinere JavaScript-Library verwendet werden um die Problemlösung zu vereinfachen.

Testing Technologie:



ABBILDUNG E.4: Wallaby Logo

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

Getestet wird die Applikation durch die von Phoenix gegebenen Testing-Tools sowie mit der Browser-Testing Library «Wallaby».

E.11 Kosten

In der Studie wurden die Projekt- sowie Betriebskosten ausgerechnet.

Der gesamte Personalaufwand beträgt **42'900** für die geplanten Stunden.

Phase	Geplante Stunden	Kosten
Initialisierung	64	9'600.- CHF
Konzept	66	9'900.- CHF
Realisierung	136	20'400.- CHF
Abschluss	64	5'400.- CHF
Total:	286	42'900.- CHF

TABELLE E.6: Projektkosten

Für die Betriebskosten wurde angenommen, dass das Produkt in der Cloud auf einer mittelgrossen Umgebung betrieben wird. Die Kosten dieser Umgebung wurde auf 150.- CHF pro Monat geschätzt.

Neben der Umgebung muss mindestens eine Domain gekauft und jährlich bezahlt werden. Die Kosten einer Domain sind rund 20.- CHF pro Jahr.

Da jediglich Open Source Software eingesetzt wird, gibt es keine Software-Lizenzen zu bezahlen.

Kostenstelle	Jährliche Kosten
Software	Keine
.com Domain	20.- CHF
Hosting	1'800.- CHF
Total:	1'820.- CHF

TABELLE E.7: Betriebskosten

E.12 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 E.8: Risiken - Schadensskala

Eintrittswahrscheinlichkeitsskala:

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

TABELLE E.9: 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 E.10: Risiken - Handlungen zur Senkung der Bewertung

E.12.1 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

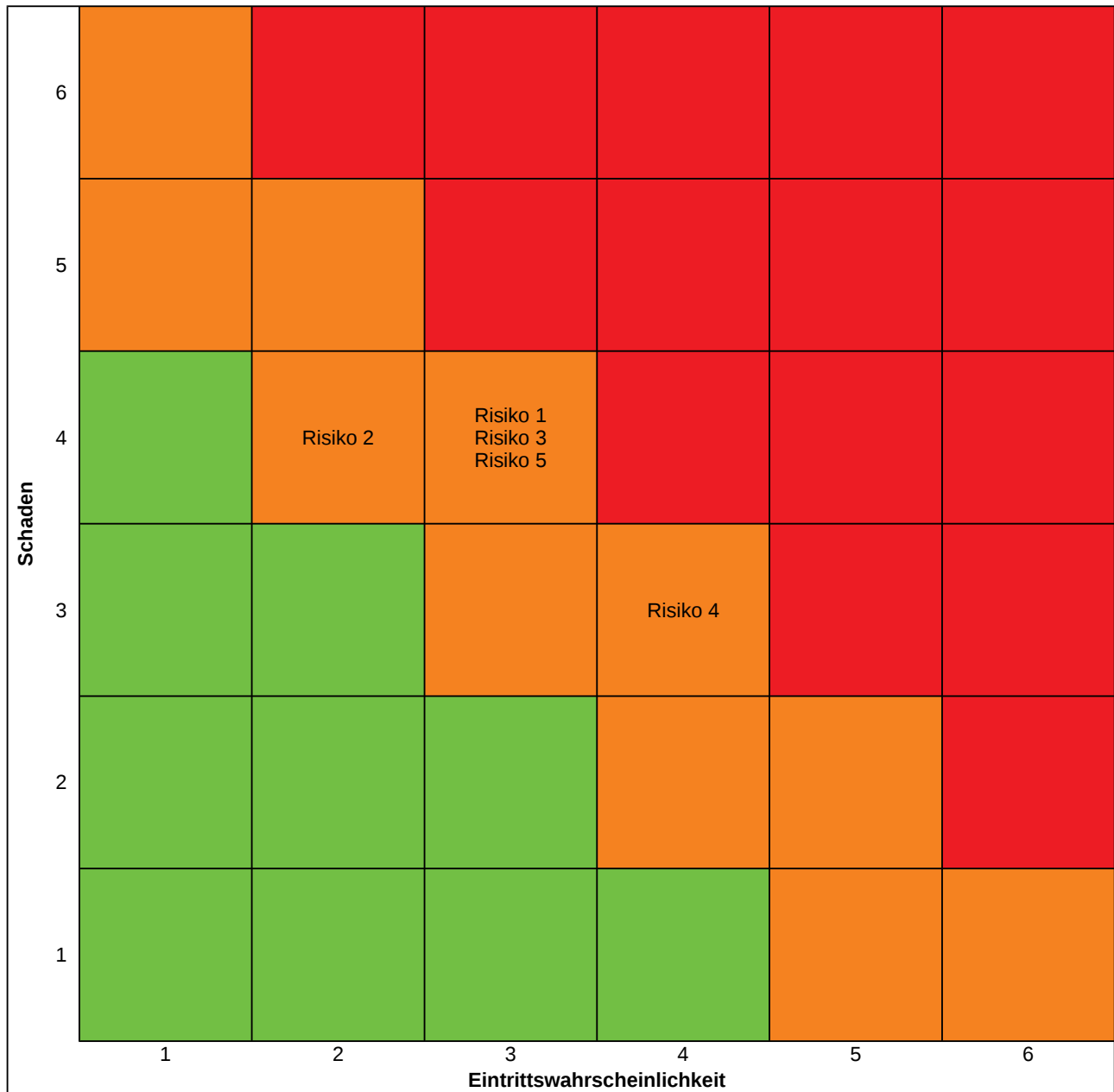
TABELLE E.11: Projektrisiken

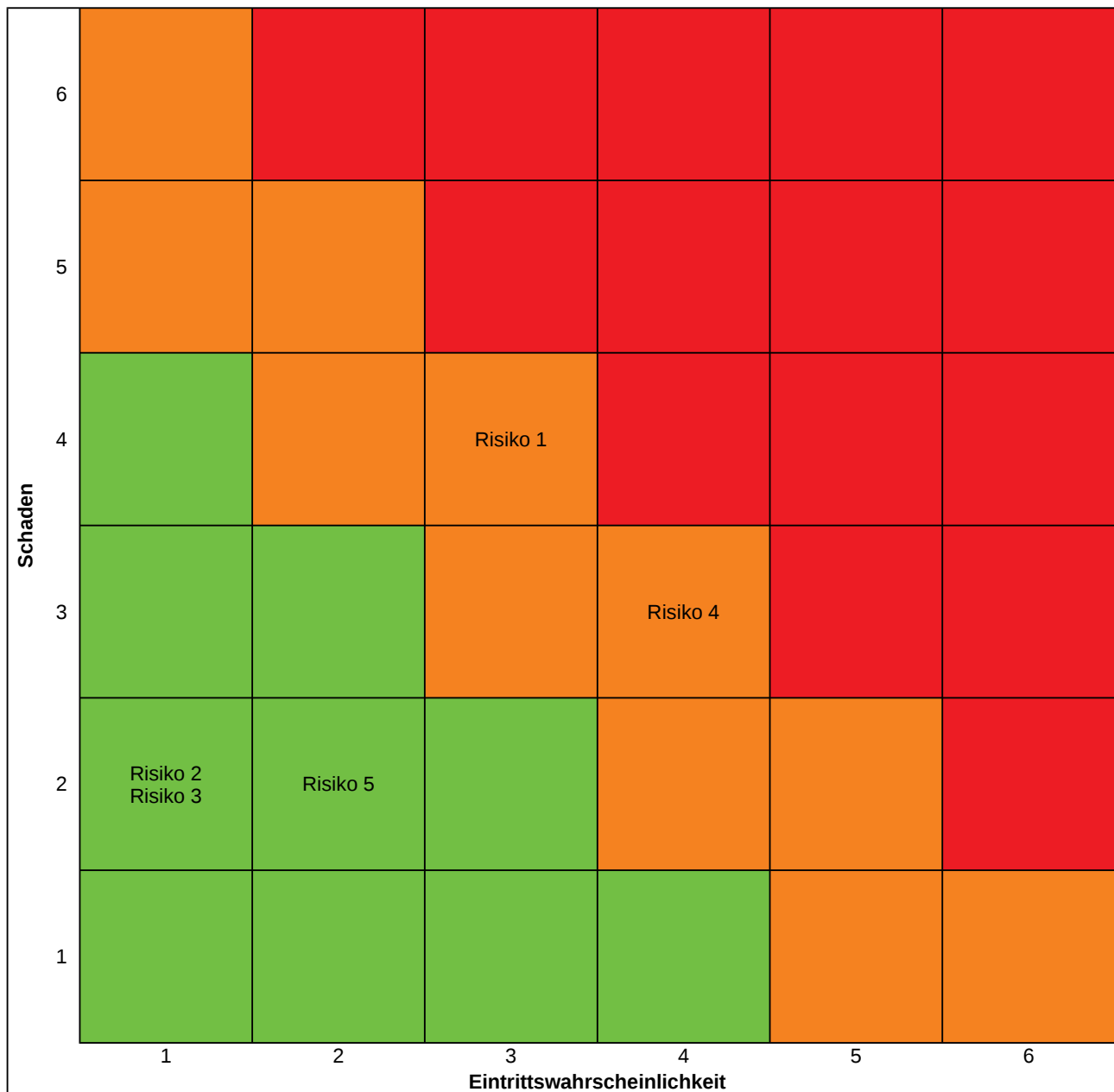
E.12.2 Massnahmen

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 E.12: Projektrisiken - Massnahmen

E.12.3 Risikodiagramm ohne Massnahmen



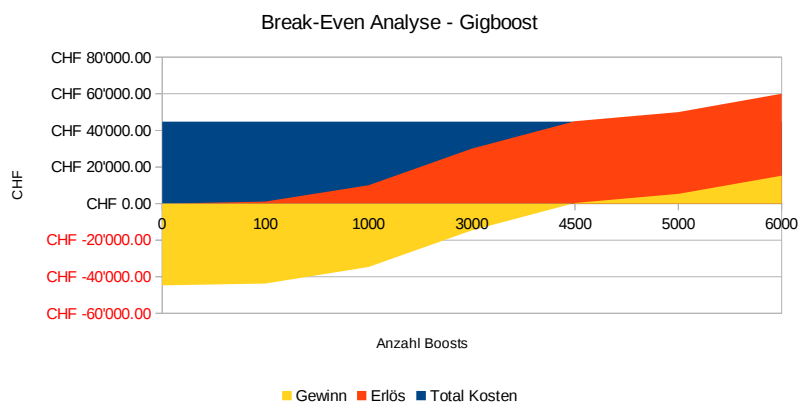
E.12.4 Risikodiagramm mit Massnahmen

Anhang F

Wirtschaftlichkeit - Gigboost

Analyse Modell – Gigboost

Anzahl Boosts	0	100	1000	3000	4500	5000	6000
Projektkosten	CHF 42'900.00	CHF 42'900.00	CHF 42'900.00	CHF 42'900.00	CHF 42'900.00	CHF 42'900.00	CHF 42'900.00
Betriebskosten	CHF 1'820.00	CHF 1'820.00	CHF 1'820.00	CHF 1'820.00	CHF 1'820.00	CHF 1'820.00	CHF 1'820.00
Total Kosten	CHF 44'720.00	CHF 44'720.00	CHF 44'720.00	CHF 44'720.00	CHF 44'720.00	CHF 44'720.00	CHF 44'720.00
„Gigboost“	CHF 10.00	CHF 10.00	CHF 10.00	CHF 10.00	CHF 10.00	CHF 10.00	CHF 10.00
Erlös	CHF 0.00	CHF 1'000.00	CHF 10'000.00	CHF 30'000.00	CHF 45'000.00	CHF 50'000.00	CHF 60'000.00
Gewinn	CHF -44'720.00	CHF -43'720.00	CHF -34'720.00	CHF -14'720.00	CHF 280.00	CHF 5'280.00	CHF 15'280.00



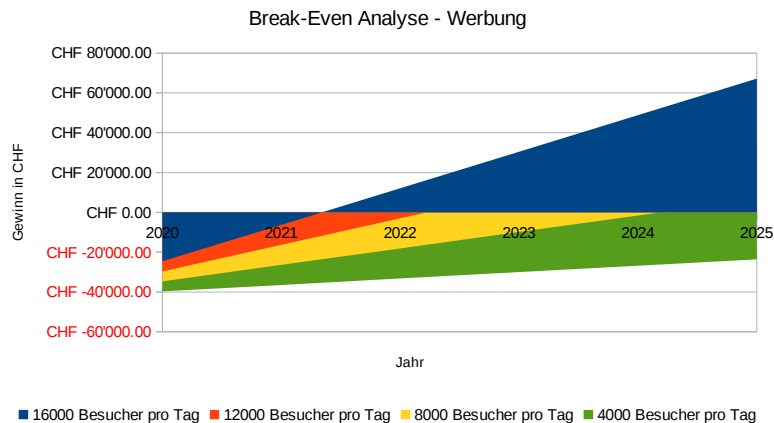
Anhang G

Wirtschaftlichkeit - Werbung

Analyse Modell – Werbung

Besucher pro Tag	2000	4000	8000	12000	16000
Besucher pro Jahr	720000	1440000	2880000	4320000	5760000
Projektkosten	CHF 42'900.00	CHF 42'900.00	CHF 42'900.00	CHF 42'900.00	CHF 42'900.00
Betriebskosten	CHF 1'820.00	CHF 1'820.00	CHF 1'820.00	CHF 1'820.00	CHF 1'820.00
Total Kosten	CHF 44'720.00	CHF 44'720.00	CHF 44'720.00	CHF 44'720.00	CHF 44'720.00
CPC	CHF 5.00	CHF 10.00	CHF 20.00	CHF 30.00	CHF 40.00
CPM	CHF 2.00	CHF 4.00	CHF 8.00	CHF 12.00	CHF 16.00
Erlös pro Monat	CHF 210.00	CHF 420.00	CHF 840.00	CHF 1'260.00	CHF 1'680.00
Erlös pro Jahr	CHF 2'520.00	CHF 5'040.00	CHF 10'080.00	CHF 15'120.00	CHF 20'160.00
Gewinn	CHF -42'200.00	CHF -39'680.00	CHF -34'640.00	CHF -29'600.00	CHF -24'560.00

	4000 Besucher pro Tag	8000 Besucher pro Tag	12000 Besucher pro Tag	16000 Besucher pro Tag
2020	CHF -39'680.00	CHF -34'640.00	CHF -29'600.00	CHF -24'560.00
2021	CHF -36'460.00	CHF -26'380.00	CHF -16'300.00	CHF -6'220.00
2022	CHF -33'240.00	CHF -18'120.00	CHF -3'000.00	CHF 12'120.00
2023	CHF -30'020.00	CHF -9'860.00	CHF 10'300.00	CHF 30'460.00
2024	CHF -26'800.00	CHF -1'600.00	CHF 23'600.00	CHF 48'800.00
2025	CHF -23'580.00	CHF 6'660.00	CHF 36'900.00	CHF 67'140.00



Anhang H

Konzept

H.1 Portalname

Der Portalname wurde in einer Brainstorming-Session von Damian Senn auf den Namen «Gigpillar» festgelegt. Der Name ist angelehnt an die Werbepfeiler in Städten, wo oft Werbeplakate für Konzerte hängen.

Die folgenden Ideen wurden in Betracht gezogen, jedoch war keine Domain mehr verfügbar oder der Name überzeugte nicht:

- upto.com («What are you up to?»)
- up-to.com
- uptoin.com
- gigup.com
- gigsta.com («Gigs to attend»)
- gigin.com
- gigin.com
- gixin.com («Gigs in»)
- dualact.com («Loud act»)
- trecnoc.com («Concert» rückwärts)

H.2 Design- und Bedienkonzept

H.2.1 Mockups

Homepage

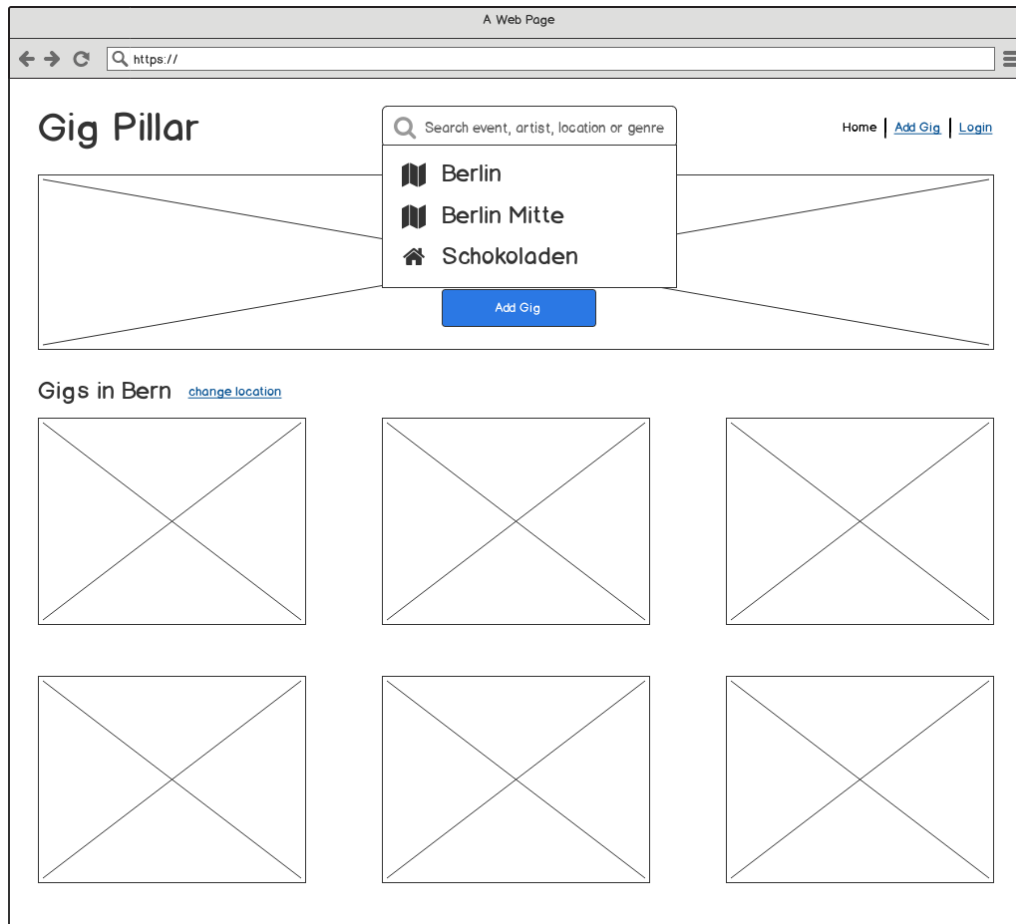
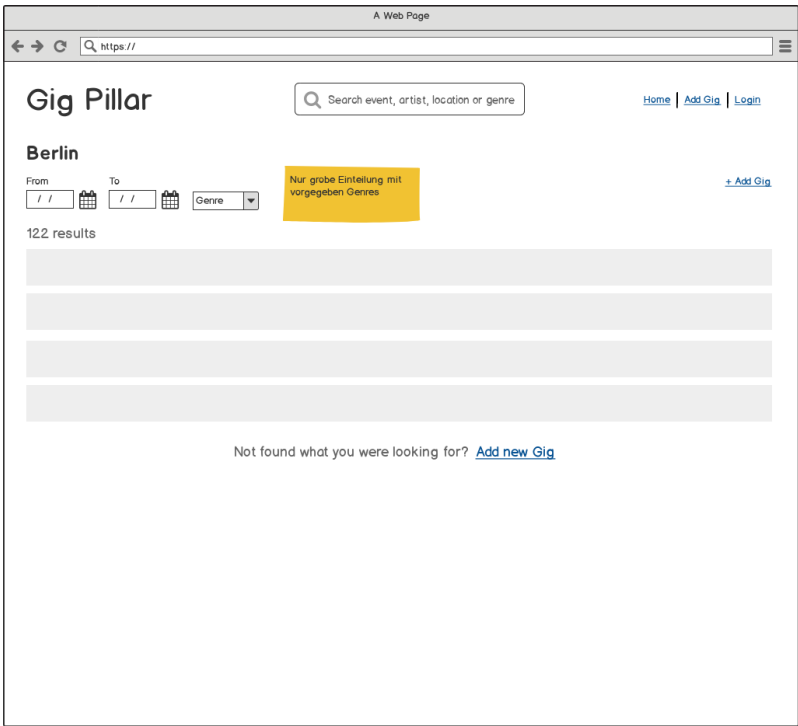


ABBILDUNG H.1: Mockup: Homepage

Suchresultate





-  **Earthless**
Schokoladen, Berlin
30. April 2019
-  **Greenfield**
Schokoladen, Berlin
31. April 2019

ABBILDUNG H.2: Mockup: Suchresultate

Gig Ansicht

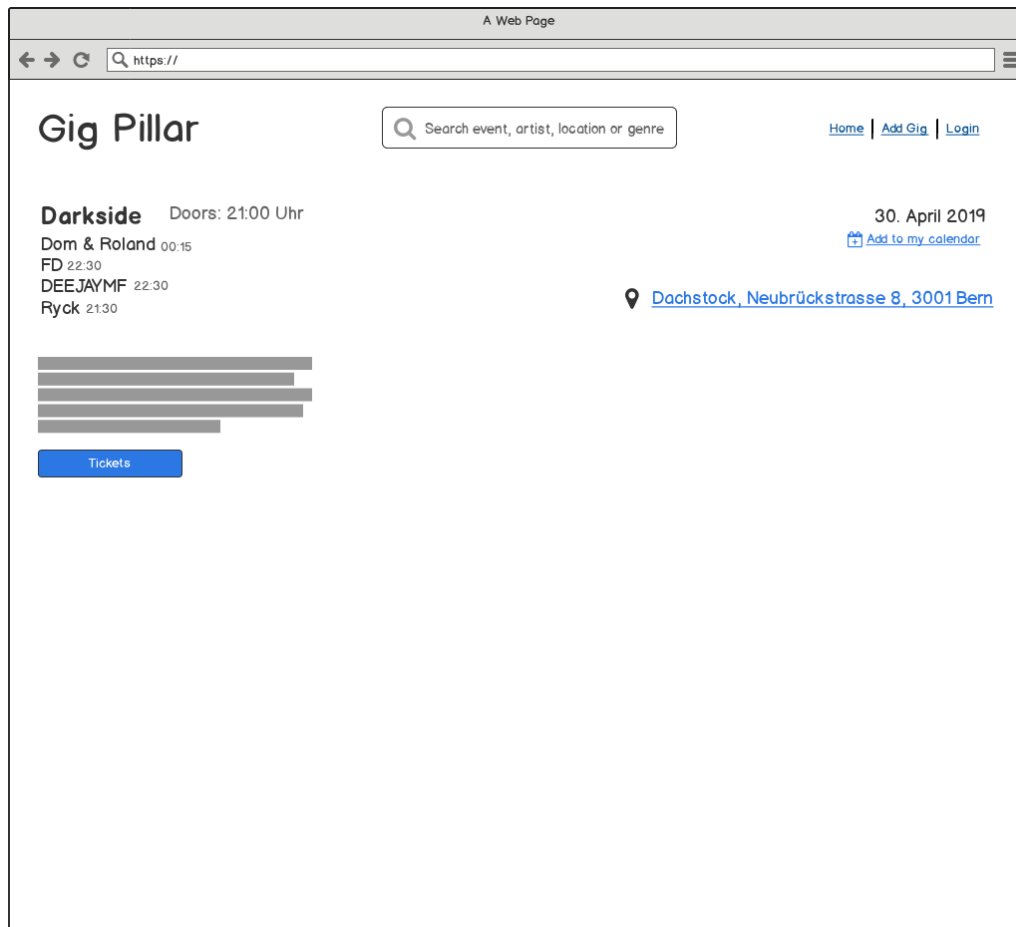


ABBILDUNG H.3: Mockup: Gig Ansicht

Gig erfassen

A Web Page

← → ↻ 🔍 https://

Gig Pillar

🔍 Search event, artist, location or genre

[Home](#) | [Add Gig](#) 👤

Add new Gig

Gig name

Where

When
 📅 Doors: [20:00](#)

Who

autocomplete

Pennywise →

☰ Pennywise at [20:00](#)

Describe Gig

Tickets

[Save Gig](#)

ABBILDUNG H.4: Mockup: Gig erfassen

Benutzerprofil

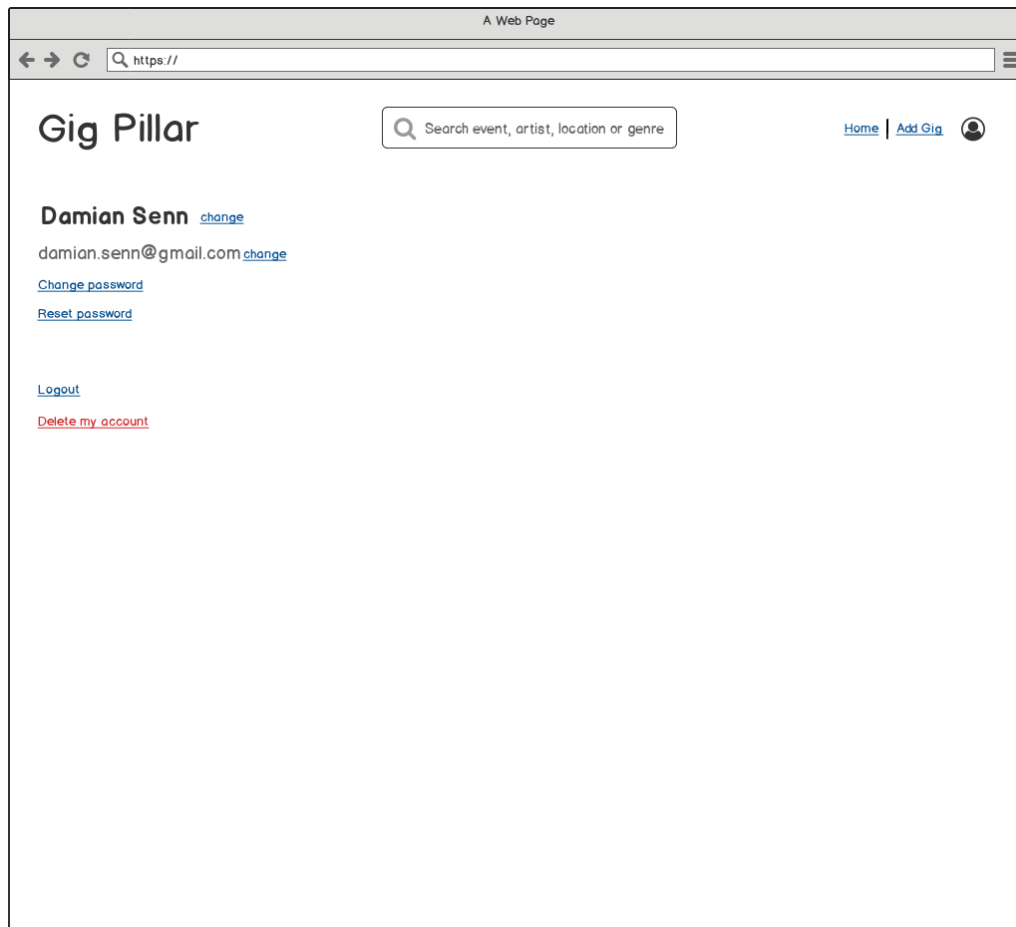


ABBILDUNG H.5: Mockup: Benutzerprofil

H.3 Softwarekonzept

H.3.1 Datenbankstruktur



ABBILDUNG H.6: Entity Relationship Diagram

H.4 Testkonzept

Anhang I

Arbeitsjournal

I.1 Sonntag 3. März

2h:

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

I.2 Dienstag 5. März

2h:

- Vorbereitung Kick-off

I.3 Mittwoch 6. März

3h:

- Vorbereitung Sitzungszimmer
- Kick-off Meeting

4h:

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

I.4 Samstag 9. März

2h:

- An Studie/Pflichtenheft arbeiten

I.5 Dienstag 12. März

2h:

- PDF Generierung und Ordnerstruktur angepasst

I.6 Samstag 16. März

3h:

- Projektplan von gantt nach ods migrieren

I.7 Dienstag 19. März

0.5h:

- Projektplan in Berichtanhang angehängt

I.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

I.9 Sonntag 31. März

5h:

- Projektziele erweitert
- Anforderungskatalog erweitert

I.10 Sonntag 31. März

8h:

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

I.11 Freitag 5. April

2h:

- An Studie weiter gearbeitet
- Variantenkriterien definiert

I.12 Samstag 6. April

4h:

- An Studie weiter gearbeitet
- Variantenbeschreibungen erstellt
- Variantenbewertungen

I.13 Mittwoch 10. April

2h:

- Brainstorming für Portalnamen

I.14 Freitag 12. April

12h:

- Screens definiert
- Mit Joshua Schär an Mockups gearbeitet

I.15 Samstag 13. April

12h:

- Risiko Management
- Variantenbeschreibungen angepasst/erweitert
- Projektauftrag

I.16 Sonntag 14. April

8h:

- Wirtschaftlichkeit
- Projektauftrag

I.17 Montag 15. April

2h:

- Initiales Datenbankschema basierend auf Mockups

Anhang J

Biweekly Reports

J.1 Kalenderwoche 11-12

Hallo Sandro, hallo Severin

Kurzes Update was in meiner Diplomarbeit in den letzten zwei Wochen gelaufen ist:

- Ein Projektplan mit mehr Detail wurde erstellt, Details für die Realisierungsphase werden sich in der Konzeptphase noch konkretisieren.
- Das Organigramm habe ich angepasst, nach dem ich mit Marc Aeby die Auftragsgeber/Projektleiter Situation abgeklärt hatte. Marc meinte, dass der Auftragsgeber nicht gleichzeitig Projektleiter sein darf. Somit habe ich wie mit Severin in Person abgemacht ihn zum neuen Auftragsgeber ernannt.
- Der Projektauftrag wurde grob strukturiert und mit dem Organigramm, Abgrenzungen, Terminplan abgefüllt. Der Projektauftrag wird im Laufe der Studie weiter ergänzt.
- Ich habe den Anforderungskatalog angefangen und Muss/Kann Kriterien definiert.
- Die Nutzerwertanalyse für den Technologie-Einsatz habe ich letzte Woche angefangen.

Weiteres Vorgehen für die nächsten zwei Wochen habe ich wie folgt geplant:

- Die Nutzwertanalyse wird fertiggestellt.
- Ein Variantenentscheid wird gefällt.
- Es wird eine Risikoanalyse erstellt.
- Aus dem obigen Output wird der Projektauftrag fertig gestellt.
- Der aktuelle Stand wird in ein PDF generiert und euch zugeschickt.

Meine Dokumente sind derzeit noch in einfachen Text-Files abgelegt, ich werde euch einen aktuellen Stand als PDF spätestens beim nächsten Report mitschicken.

Bei Fragen oder Unklarheiten stehe ich euch gerne zur Verfügung.

Gruss
Damian

J.2 Kalenderwoche 13-14

Hallo Sandro, hallo Severin

Kurzes Update was in meiner Diplomarbeit in den letzten zwei Wochen gelaufen ist:

- Ich habe den Projektplan an die korrekte HERMES 5 Struktur angepasst.
- Mit Joshua habe ich einen Termin am 12.04. abgemacht um erste Screendesigns zu erstellen.
- Die Projektziele wurden ergänzt und detaillierter beschrieben.
- Der Anforderungskatalog wurde fertig gestellt.

Aus privaten und gesundheitlichen Gründen konnte ich die letzten zwei Wochen leider nicht viel am Projekt arbeiten. Ich bin aber zuversichtlich, dass ich durch die Schulferien und dem Zürcher-Feiertag am Montag dieses Wochenende mein Defizit wieder aufarbeiten kann. Somit sieht das weitere Vorgehen leider ähnlich wie im letzten Report aus:

- Die Nutzwertanalyse wird fertiggestellt.
- Ein Variantenentscheid wird gefällt.
- Es wird eine Risikoanalyse erstellt.
- Aus dem obigen Output wird der Projektauftrag fertig gestellt.

Anbei habe ich euch den momentanen Stand meiner Dokumentation angehängt. Die Inhalte sind hauptsächlich im Anhang zu finden, da ich den Teil im Bericht ausfülle sobald die entsprechenden Anhänge fertiggestellt sind.

Bei Fragen oder Unklarheiten stehe ich euch gerne zur Verfügung.

Gruss
Damian

J.3 Kalenderwoche 15-16

Hallo Sandro, hallo Severin

Kurzes Update was in meiner Diplomarbeit in den letzten zwei Wochen gelaufen ist:

- Die Nutzwertanalysen wurde fertiggestellt.
- Die Variantenentscheide wurden gefällt.
- Die Risikoanalyse wurde fertiggestellt.
- Die Wirtschaftlichkeit wurde berechnet.
- Somit ist der Projektauftrag fertiggestellt.
- Der Produktname ist «Gigpillar».
- Es wurden Mockups für einige Screens zusammen mit Joshua erstellt.
- Basierend auf den Mockups habe ich ein erstes grobes Datenbankschema erstellt.

Das weitere Vorgehen sieht folgendermassen aus:

- Die Screendesigns im Konzept detailliert beschreiben.
- Die Software-Architektur erstellen.
- Ein Test-Konzept erstellen.
- Die Präsentation für das Zwischen-Meeting vorbereiten.
- Begin der Realisierungsphase.

Anbei habe ich euch den momentanen Stand meiner Dokumentation angehängt. Ihr findet die Studie im Anhang **D** und den Projektauftrag im Anhang **E**.

Ich freue mich euch den Stand des Projekts nächste Woche am 24.04.2019 um 09:00 an der Belpstrasse 37 vorzustellen.

Sandro ich habe dir unseren Besucherparkplatz Nummer 39 von 08:45 bis 12:00 reserviert, du findest den Anfahrtsplan im Anhang.

Bei Fragen oder Unklarheiten stehe ich euch gerne zur Verfügung.

Gruss
Damian