

Delegator		
Kenan Tabinas		
<div>delegator</div> <div></div>		
HFINF	Klasse 21b / Praktische Diplomarbeit 2025	ID: 2558

1. Management Summary

Delegator ist eine App, die speziell für wenig technikaffine Musiker entwickelt wurde. Ziel ist es, Bands eine einfache, intuitive Plattform zu bieten, mit der sie Termine, Aufgaben, Ressourcen und Kommunikation an einem Ort verwalten können. Damit soll ein Problem gelöst werden, das viele kleine Bands betrifft. Fehlende Organisation ohne Zugang zu teuren Management-Dienstleistungen.

Die Diplomarbeit umfasst die vollständige Konzeption, Entwicklung und erste Testphase der App. Sie wurde nach dem Prinzip Design First gestaltet und mit modernen Technologien wie Django (Backend), Flutter (Frontend) und PostgreSQL umgesetzt. Der Betrieb erfolgt containerisiert, ursprünglich auf einem Raspberry Pi, später auf einem MacBook, mit einem geplanten Wechsel zu einem Managed Service (z. B. Exoscale) vor dem produktiven Einsatz.

Die Zielgruppe wurde bewusst auf Bands eingeschränkt, da der Familien- und KMU-Markt als zu gesättigt eingeschätzt wurde. Die App deckt typische Bandbedürfnisse von Kalender-Synchronisation über Rollen- und Rechteverwaltung bis zu Chatfunktionen. Die technische Architektur basiert auf einer Cross-Plattform-App mit Fokus auf einfacher UX.

Eine Marktanalyse und ein Businessplan zeigen auf, dass die App mit einem Freemium-Modell wirtschaftlich betrieben werden kann. Ein Break-even ist bei realistischer Nutzergewinnung nach ca. fünf Jahren möglich. Die App wurde erfolgreich getestet. Die Grobanforderungen und Ziele wurden grösstenteils erreicht.

Die Arbeit zeigt nicht nur ein funktionierendes Produkt, sondern auch das unternehmerische Potenzial dahinter. Ob das Projekt weitergeführt wird, hängt von Ressourcen, Marktinteresse und persönlichen Prioritäten ab.

Text Generiert von Claude.

2. Inhaltsverzeichnis

Contents

1. Management Summary.....	2
2. Inhaltsverzeichnis.....	3
3. Initalisierung.....	6
3.1. Ausgangslage.....	6
3.2. Projektziele.....	7
3.3. Projektorganisation.....	7
3.4. Projektplan.....	8
3.5. Lieferergebnisse.....	9
3.6. Ressourcenplan.....	9
3.7. Risiken.....	11
3.8. Abgrenzung.....	12
3.9. Studie.....	13
3.9.1. Konkurrenzanalyse.....	13
3.9.2. Zielgruppenanalyse.....	14
3.10. Technologie.....	15
3.10.1. Varianten.....	15
3.10.2. Entscheidung.....	16
3.10.3. Bewertung.....	16
3.10.4. Fazit.....	16
3.11. Nicht Entscheidungen.....	Error! Bookmark not defined.
3.12. Service Stack.....	17
3.12.1. Mögliche Varianten.....	17
3.12.2. Break Even.....	18

3.12.3.	Fazit	18
3.13.	Wirtschaftlichkeit.....	19
3.13.1.	Konkurrenz	19
3.13.2.	Betriebskosten	19
3.13.3.	Instagram Ads	20
3.13.4.	Errechnete Neu-User	20
3.13.5.	Preismodell.....	20
3.13.6.	Breakeven	21
3.13.7.	Mögliches Risiko	21
4.	Konzept	22
4.1.	Designkonzept.....	22
4.2.	Berechtigungskonzept.....	24
4.3.	Testkonzept.....	22
4.4.	Datenschutzkonzept.....	22
4.5.	Sicherheitskonzept	24
4.6.	Datenbankkonzept	23
4.6.1.	Projekt	23
4.6.2.	User	23
5.	Realisierung.....	25
5.1.	Ausführung.....	25
5.1.1.	Backend Umgebung.....	Error! Bookmark not defined.
5.1.2.	Datenbank.....	Error! Bookmark not defined.
5.1.3.	Projekt Rechte.....	25
5.2.	Testprotokoll	26
6.	Einführung.....	29

7. Schlussbetrachtung.....	31
8. Authentizität	32
9. Anhang.....	33

Das Inhaltsverzeichnis wird für das ganze Dokument (und nicht für jeden Phasenbericht einzeln) erstellt: Titelnnummer (ca. 3 Ebenen); Titteltext; Seitennummer.

3. Vorwort

Organisation ist in vielen Lebensbereichen entscheidend – auch in der Musik. Gerade für kleinere Bands, die ohne professionelles Management auskommen müssen, stellt die Koordination von Terminen, Aufgaben und Ressourcen eine besondere Herausforderung dar. Aus eigener Erfahrung weiss ich: Kreativität gedeiht nur, wenn der organisatorische Rahmen stimmt.

Diese Diplomarbeit entstand aus dem Wunsch, genau hier anzusetzen. Ziel war es, eine App zu entwickeln, die wenig technikaffine Musikerinnen und Musiker im Alltag unterstützt – pragmatisch, intuitiv und ohne überladene Funktionen. Das Projekt verbindet technisches Know-how mit persönlicher Motivation und zeigt, wie aus einer Idee ein funktionierendes Produkt entstehen kann.

Natürlich lief nicht alles nach Plan. Wie sagt man so schön: Wenn ein Plan reibungslos funktioniert, war es kein IT-Projekt.

Text generiert von Claude.

4. Initialisierung

4.1. Ausgangslage

Das Leben in einer Band ist schwierig. Gerne möchte man vor allem Zeit in kreative Prozesse stecken. Grosse Bands haben einen Manager, der solche Aufgaben übernimmt. Der nimmt jedoch gerne 15% des Umsatzes. Leider sind kleine Bands weit von grünen Zahlen entfernt. Dazu kommt noch, dass eine kleine Band ohne Organisation klein bleibt.

Es existieren bereits Tools, welche in diese Richtung gehen. Diese sind aber eher für kleinere Unternehmen und Technik affine Menschen. Daher haben wollen wir eine Lösung für Technikuninteressierte Musiker. Nach dem Motto «Weniger Admin mehr Musik». Eine Band ohne Organisation funktioniert nicht. Wenn wir schon nicht darauf verzichten können, soll es wenigstens so einfach wie möglich sein.

Musiker wollen nicht jedes Mal, um Termine zu sehen, in eine andere App wechseln müssen. Musiker wollen nicht für Dateien, Chats, Aufgaben und Kalender vier Separate Apps benutzen. Familien Mitglieder sollen Zugriff auf Kalender haben und über alles Wichtige informiert werden, ohne von unnötigen Informationen bombardiert zu werden.

Gäbe es doch nur etwas, welche diese Probleme zielgruppengerecht lösen könnte.

4.2. Projektziele

Folgende Ziele sollen bis Ende Projekt erreicht werden.

ID	Ziel	Beschreibung
Z001	Projekt	Ich kann Projekte erstellen
Z002	Aufgaben	Ich kann verschiedene Aufgaben in einem Projekt erfassen
Z003	Template	Ein Projekt Template erstellt mir Aufgaben
Z004	Termin	Ich kann ein Termin bei der Aufgabe definieren
Z005	Zuweisung	Ich kann Aufgaben einem User zuweisen
Z006	Kalender 01	Ich kann pro User ein Aufgaben Kalender definieren
Z007	Kalender 02	Ich kann auf einem Kalender der Zeit Rahmen des ganzen Projektes sehen
Z008	Kalender 03	Ich kann auf einem Kalender den Zeitrahmen der ganzen Unternehmung sehen
Z009	Kalender 04	Ich kann auf einem Kalender die Ressourcen eines Users bearbeiten
Z010	Kalender 05	Ich kann auf meiner Kalender APP meinen Kalender synchronisieren

Tabelle 1: Ziele

4.3. Projektorganisation

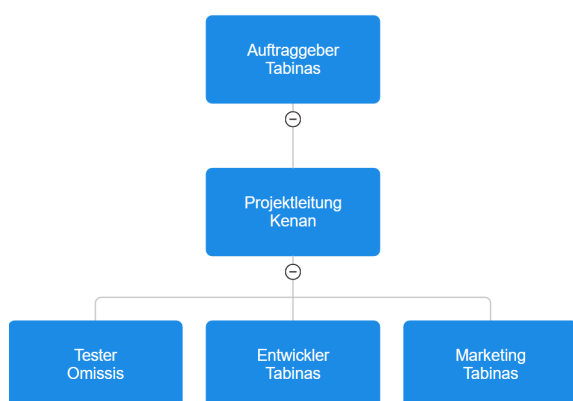
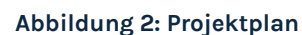


Abbildung 1: Projekt Organisation



4.5. Liefserergebnisse

Aufzeigen der Ergebnisse, die während der Diplomarbeit erarbeitet werden bzw. wurden und einen massgeblichen Beitrag zur Erfüllung der Projektziele leisten.

ID	Ziel	Beschreibung
L001	Projekt	Ich kann Projekte erstellen
L002	Kanban Board	Ich kann verschiedene Aufgaben in einem Projekt erfassen
L003	Kalender	Ein Projekt Template erstellt mir Aufgaben

Tabelle 2: Lieferergebnisse

4.6. Ressourcenplan

4.6.1. Externe Kosten

Die externen Kosten wurden auf 0 geschätzt.

4.6.2. Interne Kosten

Basierend auf dem IST Zustand des Zeitplanes. Bei 120 CHF pro Stunde.

Phase	Zeit in h	Geplant in CHF
Initialisierung	100	12000
Konzept	32	3840
Realisierung	80	9600
Einführung	32	3840
Abschluss	15	1800
Total	259	31080

Tabelle 3: Interne Kostenschätzung

4.6.3. Umgebung

Damit wir die nötigen Ressourcen verstehen müssen wir die Architektur verstehen.

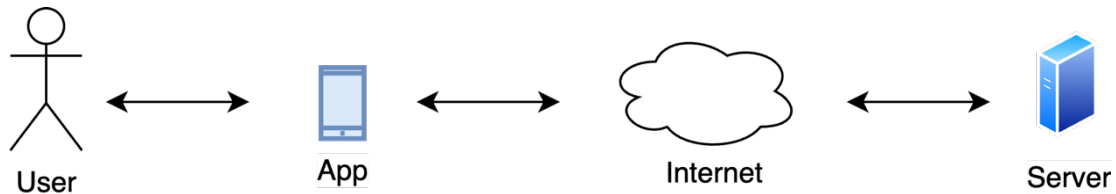


Abbildung 3: Einfache Darstellung

Am Anfang der Entwicklung lief der Server auf dem Raspberry Pi 5. Im Verlauf des Projektes bin ich auf ein M4 Pro Mac-Book gewechselt. Das System bleibt aber das gleiche.

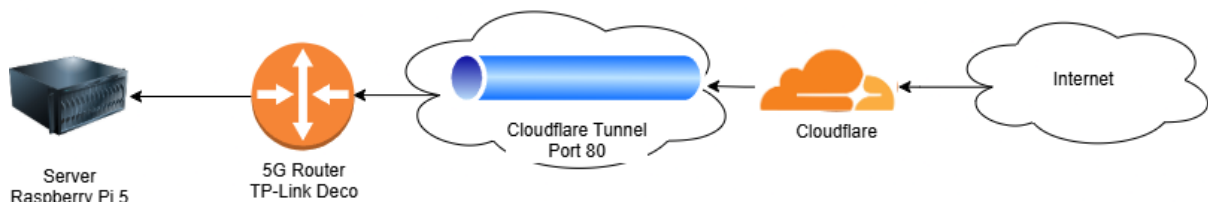


Abbildung 4: Server zu Internet

Es wurde sich zu diesem Zeitpunkt, keine Gedanken zu den Kosten der Umgebung gemacht.

Es laufen fünf verschiedene Container die je einen Zweck erfüllen.



Abbildung 5: Darstellung der Container

4.7. Risiken

Bei der folgenden Auflistung gibt es einige der möglichen Risiken, um sich auf das Wesentliche zu begrenzen.

NR	Beschreibung	Eintritts- wahrscheinlichkeit	Auswirkung	Risiko
1	Backend System Ausfall	2	2	4
2	Cyber Angriffe	3	3	9
3	Fehler in der Mobileapp	2	3	6
4	Ausfall von Plattform Services	2	2	3
5	Konkurrenzverhalten	1	3	6
6	Datenschutzverletzung	2	2	4

Tabelle 4: Risiko Analyse

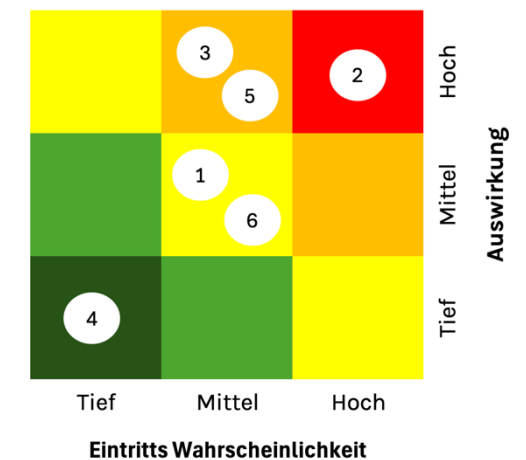


Abbildung 6: Risiko Matrix

4.8. Abgrenzung

In der Folgenden Grafik wird die Grenze des Projektes visualisiert. Alle Elemente im grünen Zylinder sind Teil vom Projekt. Der Fokus lag auf Sicherheit, der Userexperience, Design, Konzepten und einem proof of concept. Der tatsächliche Release der App und Vermarktung, kommt nicht nur mit einem hohen Zeitaufwand, sondern auch mit einem finanziellen Risiko einher welches, während dem Projekt nicht eingegangen wurde.

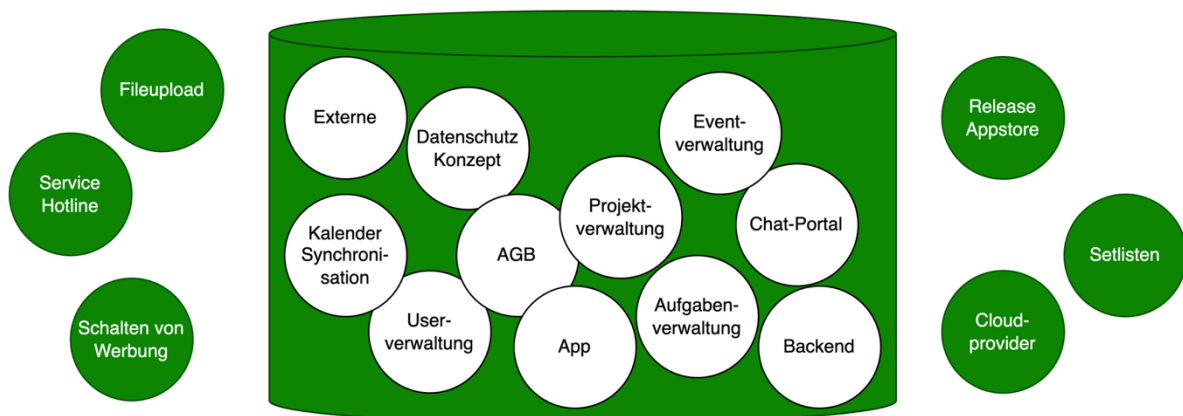


Abbildung 7: Abgrenzung

4.9. Studie

Im Folgenden wird nur auf die wesentlichen Teile der Studie eingegangen.

4.9.1. Konkurrenzanalyse

Die Frage, die sich am Anfang stellt, steige ich bereits in einen übersättigten Markt ein oder ist noch was zu holen. Bei der Analyse hat sich ergeben, dass es bereits etablierte Player auf dem Markt gibt. Hier folgt eine stark heruntergebrochene Zusammenfassung und das Fazit davon.

App	Preis	Feature Abdeckung
Trello	Kostenlos (begrenzte Features) 5 CHF pro User pro Jahr	Mittel
Asana	Kostenlos (bis 15 User) 11 CHF pro User pro Jahr	Hoch
Clickup	Kostenlos (begrenzte Features) 7 CHF pro User pro Jahr	Hoch
Google Calender	Kostenlos	Tief

Tabelle 5

Asana und Clickup erschienen als Dorn im Auge. Jedoch ist Clickup komplett überladen. Die Einstellungsmöglichkeiten scheinen schier unendlich. Das mag zuerst wie ein Vorteil klingen, jedoch ist es auch ein Nachteil. Daraus haben sich folgende Erkenntnisse ergeben:

- Positionierung: Informatik Uninteressierte haben andere Bedürfnisse als der Entwickler. Die Applikation muss mit Einfachheit herausstechen.
- Die Zielgruppe «kleine Unternehmen» ist bei dieser Konkurrenz komplett aus dem Rennen.

4.9.2. Zielgruppenanalyse

Bei der Vorstellung des Projektes war die Primäre Zielgruppe Musiker. Marc Abby und Ralph Maurer sprachen davon, dass dieses Produkt einen Nerv triffe. Dieser wird auch für Familien und kleinere Unternehmen interessant sein.

Diese Anmerkungen haben sich in Studie wieder gefunden. Ohne auf alle Details einzugehen hat sich durch die Zielgruppen Analyse folgendes ergeben.

In der Studie stehen Quellen und Rechenwege für folgende Berechnungen. Diese Zahlen beschränken sich auf den DACH-Raum (Deutschland, Österreich und Schweiz)

Zielgruppe	Möglicher Markt
Bands	ca. 148 000 Bands
Familien	ca. 39 000 000 Mehrpersonen Haushalte

Tabelle 6

Auf den ersten Blick sind Familien. Respektive Mehrpersonen Haushalte, als Zielgruppe interessanter. Bei einer weiteren Analyse hat sich aber folgendes ergeben:

- Die Zielgruppe «Familien» ist 24-mal grösser.
- Es gibt bereits über 10 Apps im Familien Bereich.
- Die App OurHome ist kostenlos und hat eine hohe Features Abdeckung.
- Im Band Bereich gibt es nur 2 mit tiefer Feature Abdeckung

Folgende Entscheidung wurde getroffen:

Zielgruppe	Hinweis
Bands	Die Zielgruppe der App
Familien	Übersättigter Markt
Kleine Unternehmen	Ist bei der Konkurrenz Analyse herausgeflogen

Tabelle 7

4.10. Technologie

In der Studie musste noch geklärt werden, welchen Technologie Stack gefahren wird. Dank der beiden vorgängigen Analyse ergaben sich Punkte, die bei der Entscheidung massgebend waren.

Folgende Punkte wurden beachtet:

- Die Userexperience muss überzeugen
- User möchten nicht das Notebook zücken, um ein Konzert einzutragen

4.10.1. Varianten

Folgende Technologien Varianten standen zur Auswahl:

- V1: Web (keine App)
- V2: Web-Viewer-App
- V3: X-Plattform App
- V4: Native App

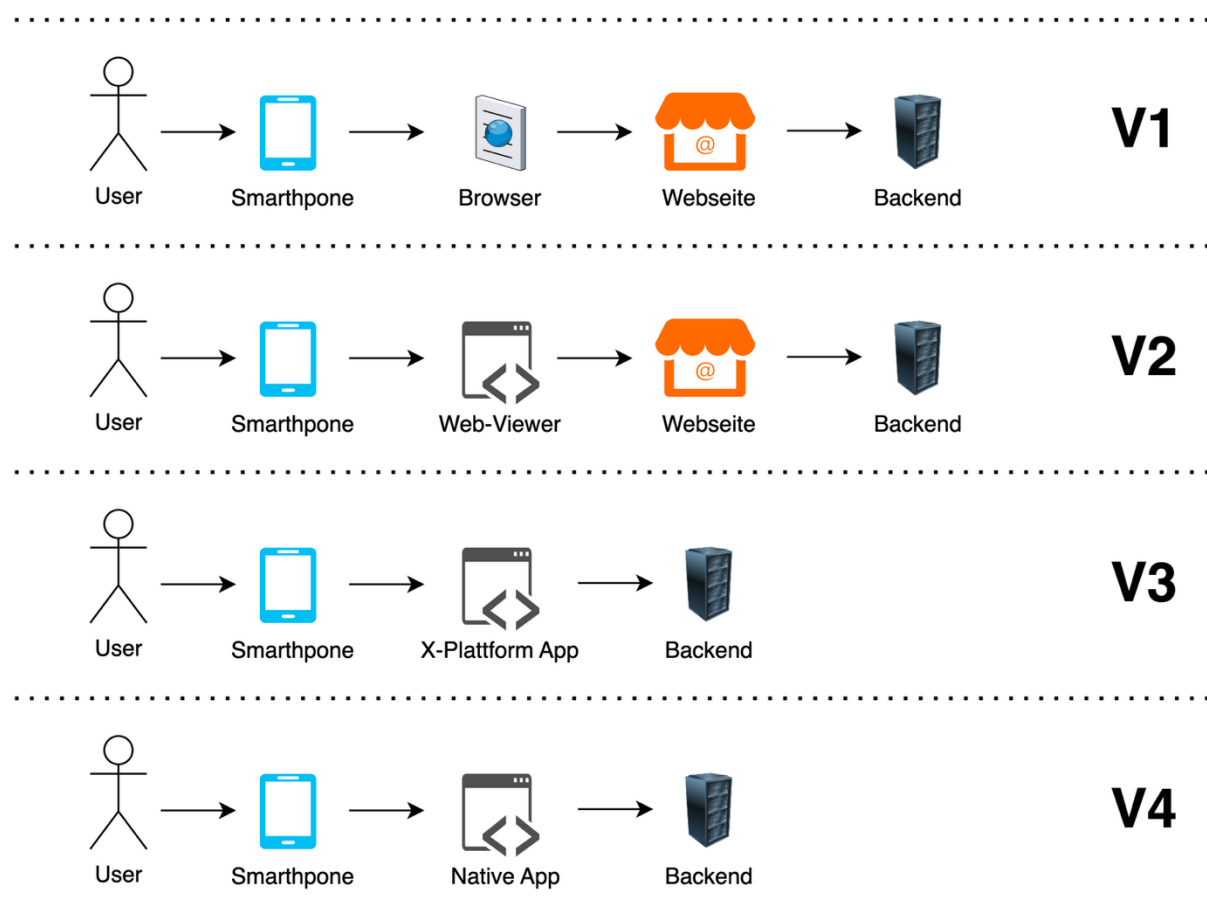


Abbildung 8: Technologievarianten

4.10.2. Entscheidung

Der gesamte Variantentscheid befindet sich im Anhang Studie. Um die Entscheidung nachvollziehen zu können muss man wissen, dass Web-View Apps zwar schnell in der Entwicklung sind und Vorteile der Nativen Entwicklung haben, jedoch verstossen sie etwa gegen Appstore Richtlinien vom Goolge Playstore und Apple Appstore. Zusammenfassend kann man sagen wegweisend waren die vorgängigen Analysen. Diese haben der Userexperience einen hohen Stellenwert zugesprochen.

4.10.3. Bewertung

Alle Kriterien werden von 1 bis 3 bewertet. Wobei 3 am besten ist und 1 am schlechtesten.

Kriterium	WebOnly	WebView	Native	X-Plattfom
Entwicklung	3	3	1	2
Updates	3	3	1	3
Appstore	1	2	3	3
UX	1	2	3	3
Total	8	9	8	10

Tabelle 8: B2 Bewertung

4.10.4. Fazit

Die WebOnly ist wegen der fehlenden App rausgeflogen. Bandmitglieder möchten nicht jedes Mal das Notebook aufklappen, um zu sehen, wann die nächste Probe ist.

Die Native App ist wegen dem Entwicklungsaufwand rausgeflogen. Es braucht zu viel Zeit und Geld mehrere Apps zu Entwickeln und zu Unterhalten.

Bei genauerem Hinsehen sind die WebViewApp und die Cross-Plattform nahe beieinander. Für die Entwicklung von WebView Apps bietet sich die Verwendung von Cross-Plattform-Framework an.

Die User Experience liefert die finale Entscheidung. Die User Experience auf einer Browser ähnlichen App ist nicht dieselbe wie bei einer Nativen App. Somit können wir die UX und Qualität steigern, indem wir uns auf ein Segment limitieren.

4.11. Deployment Architektur

In der Studie wurden folgende Deployment-Architekturen (CD/CI) und Anwendungs-Architekturen verglichen. Im Wesentlichen ging aus der Nutzwertanalyse hervor, dass sich **Hybridarchitektur** mit **Rolling Deployment** am besten für das Projekt eignet. Stark gewichtet wurden die Entwicklungskosten und Betriebskosten, da es sich um ein Ein-Mann-Projekt handelt.

Architektur	Beschreibung
Monolith	Alle Komponenten in einem einzigen Deployment-Paket.
Microservices	Aufgeteilt in Services (User-Service, Project-Service usw.)
Orchestrierung	Container-Lösung mit Kubernetes Management
Serverless	Function-as-a-Service (Cloud-Provider)
Hybrid	Monolith mit klarer modularer Trennung, Container-basiert

Tabelle 9: Anwendungs-Architekturen

Deployment	Beschreibung
Blue-Green	Zwei Umgebungen welche sich Abtauschen
Rolling	Mehrere Container werden schrittweise geupdated
Canary	Zwei Container, z.B: 5%, läuft bereits auf Version B.

Tabelle 10: Deployment-Architekturen

4.12. Service Stack

Wie bereits in der Abgrenzung illustriert wurde, ist die Bereitstellung des Services nicht ein Teil der Arbeit. Trotzdem wurden dazu ausführlich Gedanken gemacht. Diese hier aufzulisten, würde den Rahmen sprengen.

Es wurde ausgiebig verschiedene Provider wie auch Hardwarekombinationen angeschaut. Vom High-Tech Serverrack mit Rauchmelder bis zu managend all-in-one Lösung war alles dabei.

4.12.1. Mögliche Varianten

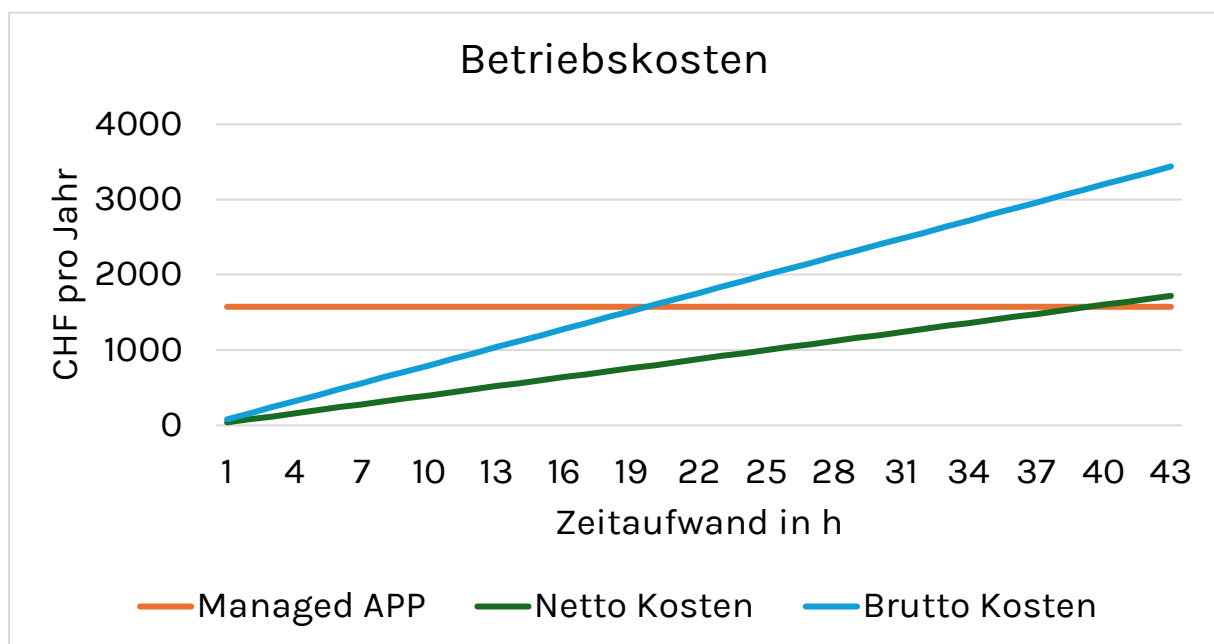
Variante	Bezeichnung
V1	On-Premise
V2	VPS
V3	CaaS

Tabelle 11: Service Stack

4.12.2. Break Even

Die Berechnung basierte auf folgendem Gedanken. Es besteht die Möglichkeit, dass ich dieses Projekt ausserhalb der Diplomarbeit weiterverfolge. Nehmen wir an ich bleibe zu 80% festangestellt. Die anderen 20% investiere ich in den Betrieb und Weiterentwicklung dieser Software.

Wenn ich mit meinem Lohn rechne, bin ich bei 40 CHF pro Stunde. Wenn ich aber Abzüge selbst bezahlen muss bin ich eher bei 80 CHF pro Stunde.



4.12.3. Fazit

Bisher war ich davon überzeugt, dass nichts günstiger ist als On-Premise. Da ein Managed Container bei 1500 CHF pro Jahr anfängt. Wenn man aber einrechnet, dass ich auch alles selbst machen muss. Logs, Backup, Sicherheit usw. Darf ich nur 20 - 40h Pro Jahr aufwenden, um noch günstiger zu sein. In Studie habe ich mehrere Szenarien aufgelistet, wie der Support meiner eigenen Software aussehen könnte. 20h im Jahr ist utopisch. Daher ist die Entscheidung klar auf einen Managed Service gefallen.

4.13. Wirtschaftlichkeit

4.13.1. Konkurrenz

Damit wir Zahlen zur Berechnung haben, habe ich die Preise der Konkurrenz angeschaut. Dabei ergaben sich die 7 CHF pro Monat

Model	Prei pro Monat in CHF	Preis im Jahr in CHF	Anzahl Kunden für Betriebskosten
CoziFamily	5	60	27
Band Helper Pro	9	108	15
Schnitt	7	84	19

Tabelle 12

4.13.2. Betriebskosten

In der Studie ist detailliert aufgelistet und beschrieben wie es zu folgenden Zahlen gekommen ist. Relevant für das Verständnis, sind die 10% für das Marketing. Diese kommen aus einer Empfehlung für startende Unternehmen.

Folgende Zahlen beziffern die Kosten der Software im ersten Jahr.

Position	CHF	%
Diplomarbeit	16400	29
Aufwand	33280	59
Marketingbudget	6000	10
Server	1570	2
Total	57250	100

Tabelle 13

4.13.3. Instagram Ads

Im Folgenden ist eine Aufstellung basierend auf dem neuen Budget von 10%. Leider gibt es 16 CHF nicht als Tages Rate bei Instagram. Daher habe ich mich für 18 CHF entschieden. Daher ist das Budget auf 11% gestiegen.

Budget pro Jahr (CHF)	Budget pro Tag (CHF)	User Reach (Min)	User Reach (Max)
6570	18	1200	3200

4.13.4. Errechnete Neu-User

In der Studie ist genau erklärt, wie es zu folgenden Zahlen kommt. Es gibt noch andere Plattformen als Instagram. Diese Berechnung gibt uns ein Gefühl dafür, wie lange es dauert.

Case	Reach pro Jahr	Clicks	Neue User
Worst	438'000	6570	131
Best	1'168'000	23360	467

Tabelle 14: Instagram CPA bei 11%

4.13.5. Preismodell

Der Gedanke war, der User hat die Möglichkeit die App zu Testen und zu schauen, wie sich die App anfühlt. Das Preismodell bietet einen Rahmen, ohne alle Features gleich weg zu geben. Es ist ein Premium Feature für welttourende Bands, welche eine hohen Wert auf Sicherheit legen. Diese würden auf einem separaten Server einen Zugriff erhalten.

Allein Benutzen Kostenlos	Zusammen pro Jahr 72 CHF pro Band (Ein Monat Rabatt)	Zusammen pro Monat 7 CHF pro Band
------------------------------	------------------------------------------------------------	--------------------------------------

Tabelle 15: Preimodell

4.13.6. Breakeven

Basierend auf den Betriebskosten, dem Marketingplan und dem Preismodell besteht die Möglichkeit das die Applikation innerhalb von 5 Jahren grüne Zahlen schreibt.

Jahr	Kosten (TCHF)	User	Gewinn (TCHF)	Saldo
1	58	131	-48	-48
2	41	263	-22	-71
3	41	394	-13	-36
4	41	526	-4	-17
5	41	657	6	2
6	41	788	15	21
7	41	920	25	40

Tabelle 16: Breakeven Studie

4.13.7. Mögliches Risiko

Nehmen wir an, ich rechne meine Zeit nicht mit ein. Ich rechne nur Kosten, welche ich an für Werbung und Server zahle. Im Worstcase bezahle ich im ersten Jahr 7000 CHF ohne etwas zurückzubekommen.

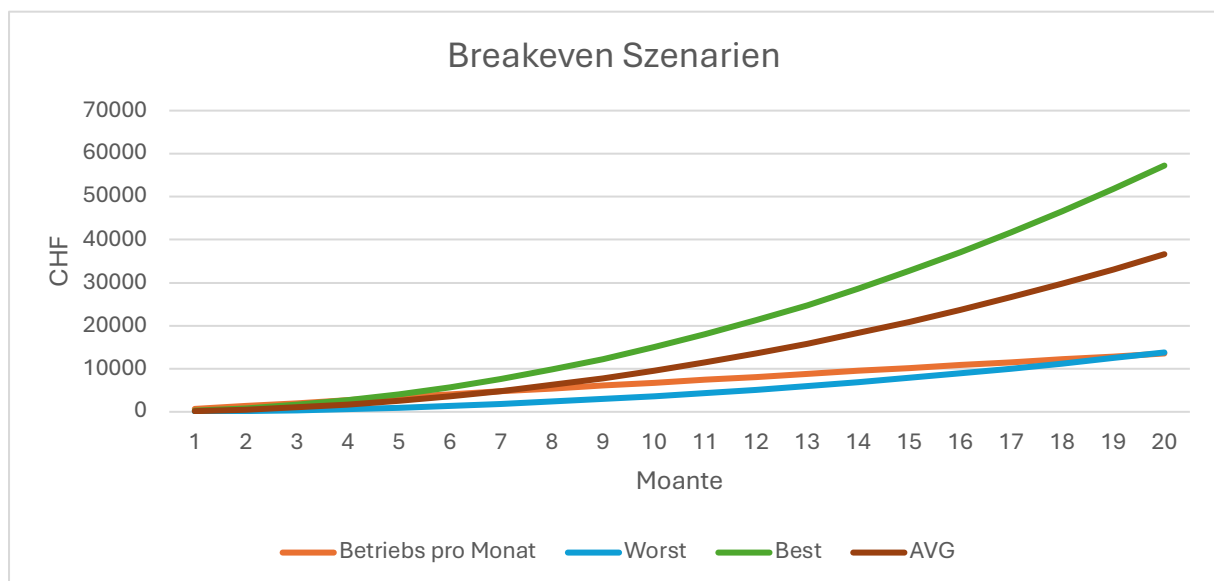


Abbildung 9: Breakeven Kosten

5. Konzept

5.1. Designkonzept

Der Design Prozess ist vollständig dokumentiert und im Anhang zu finden. Folgende drei Punkte sind relevant.

Es wurden alle möglichen Features, welche die App einmal haben könnte, geplant und haben einen festen Platz im Design bekommen. Welche Features umgesetzt wurden folgt später im Realisierung Teil.

Es wurde ein «Design First» Ansatz gewählt. Das Design war der erste Schritt und hat alle zukünftigen Schritte definiert.

Die Wireframes aus Design haben sehr bei der Entwicklung geholfen. Positiv anzumerken ist, wie nahe dem Endresultat am Design ist. Darauf wird aber noch später im Realisierung Part eingegangen.

5.2. Datenschutzkonzept

Für die Planung des Releases wurde ein Datenschutz Konzept erstellt. Dieses ist im Anhang zu finden ist. Die Wesentliche Punkte daraus sind:

- Betroffene haben Rechte (Einsicht, Deaktivierung und Vergessenwerden)
- US-Cloud Prodivder sind nach Schermen II nicht Rechtskonform.
- Ich habe eine Verantwortung und hafte auch dafür.
- Es braucht eine Roadmap, damit es nicht verloren geht.

5.3. Testkonzept

Für das Testen wurde ein Testkonzept erstellt. Dieses ist im Anhang zu finden ist. Die Wesentliche Punkte sind die Verschiedenen Teststrategien.

Nr.	Teststrategie	Beschreibung
01	Contract Testing	Test der API mit Postman
02	Integrationstest	Verknüpfung von Frontend und Backend mit Flutter-Unit-Tests
03	Anforderungsabdeckung	Testfälle basieren auf Anforderungen aus der Initialisierungsphase
04	Usability-Test	Ungeübte Benutzer*innen führen Use-Cases ohne Anleitung aus
05	Security-Test	Überprüfung der Endpoints auf Schwachstellen

Tabelle 17: Teststrategien

5.4. Datenbankkonzept

Im Anhang gibt es ein Marmaid generiertes Diagramm von Claude. Das ist aber keine Hilfe bei der Verständlichkeit. Um die Tabellen zu verstehen, beschränken wir uns auf das Wichtigste und unterteilen sie in zwei Kategorien.

5.4.1. Projekt

Der Kern der Datenbank ist das Projekt. Alle Detailinträge sind modular und besitzen die Projekt_Id. So braucht es keinen Typ. Das Generalisieren / Spezialisieren wurde so umgangen.

Tabellennamen	Hinweis
Setlist	Song und Reihenfolge.
Timetable	Ein einzelner Eintrag z.B. 17:00 Ausladen.
Task	Kann beliebig viele zum Projekt hinzugefügt werden.
Chat	Wird automatisch erstellt, sobald das Projekt erstellt wird.
Recording	Ist eine einzelne Audio-Datei.
Mood	Ein Mood ist ein Eintrag in einem Moodboard. Also wie eine Impression in Form von meistens Fotos.

Tabelle 18: Projekt Datenbank

5.4.2. User

Die Zweite Hälfte sind die User. Die folgenden Tabellen sind auch per 1-n verbunden.

Tabellennamen	Hinweis
Externe	User welche zu einzelnen Projekten hinzugefügt wurden
UserOrganisation	Zwischentabelle zwischen User und Organisation
Message	Jeder Message hat auch einen User
History	Jede Aktivität hat auch einen User
Task	Jede Aufgabe kann einem User zugewiesen werden

Tabelle 19: User Datenbank

5.5. Sicherheitskonzept

Im Anhang wird im Dokument «Design by Security» ausführlich darauf eingegangen. Im Wesentlichen ergab es Konkrete Massnahmen, geplante Massnahmen und einige Aha Momente:

- Bei Gewisse Massnahmen reicht die Verwendung von Middleware
- Nur auf Cloud Provider setzen verstösst gegen 3-2-1-1-0

5.5.1. Konkret Dokumentierte Massnahmen

Layer	Massnahmen
APP	Certificate Pinning (via Middleware) Flutter Secure Storage (via Middleware) Input Validation (via Regex)
Cloudflared	DDoS Protection (automatisch) Bot Management (automatisch) Rate Limiting (automatisch)
Django	CSRF Protection (via Middleware) SQL Injection Prevention (via Django ORM) Secure Cookie (via Middleware)
PostgresSQL	Field-level encryption (für z.B. Passwörter) Service User mit Least Privilege

Tabelle 20

5.5.2. Geplante Massnahmen gekürzt

Security Controls	Geplante Umsetzung
Angriffsfläche minimieren	Cloudflared Zero Trust Tunnel ermöglicht Verbindung, ohne einen einzigen Port zu öffnen.
Verschlüsselung	Inter-Container Verschlüsselung mit z.B. Isito
Authentifizierung	MFA für Logins Risk-based Auth (Standort-Anomalien, Gräte Profiling) z.B. loginradius
Monitoring	Zentralisiertes Log mit z.B: ELK Stack
Compliance	Externer Penntest ins Budget einfliessen lassen Jährliche Audits

Tabelle 21: Security Controls

6. Realisierung

6.1. Ausführung

Das Projekt ist nicht ohne Rückschritte passiert. Zu einem Zeitpunkt wurde 2 Tage Arbeit gelöscht. Nicht weil kein Backup existierte, sondern weil mit sämtlichem Hilfsmittel es nicht möglich war einen Fehler zu finden. Das Animierte zu einer Professionellen Methodik, welche verhalf reibungsfrei am Projekt weiterzuentwickeln. Im Folgenden wird auf die Berechtigungen und die drei Hürden welche, zu drei grossen Aha-Momente geführt haben.

6.1.1. Projekt Rechte

Was am Anfang zur Verwirrung beigetragen hat, ist die Tabelle UserProject. Man könnte meinen diese listet alle Projektteilnehmer auf. Jedoch wurde sich während der Design Phase gegen eine kompliziertes Rollen- und Berechtigungssystem entschieden. Lustigerweise ist auch eines daraus geworden, jedoch nicht für den User.

Wenn alle Band-Mitglieder zu jedem Projekt hinzugefügt werden müssen, füllt es die Tabelle UserProjekt mit immer gleichen Einträgen. Diesen Overhead wurde durch ein anderes System ersetzt.

Alle User werden via Tabelle User Organisation der Organisation hinzugefügt. Sie können mit 5 Verschiedenen Rollen hinzugefügt werden. Sobald sie Teil der Organisation sind, erhalten sie automatische Zugriffe.

ID	Rolle	Automatischer Zugriff
1	Admin	Alles
2	Long-Term member	Projekte, Verträge, Geld, User Verwaltung
3	Member	Projekte
4	Familie	Kalender Synchronisation
5	Fans	Zugriff auf öffentliche Kalender
6	Externer	Keine (Mixer*in, Videograph*in, Photograph*in.)

Tabelle 22: Zugriff

Die Tabelle UserProject ist nur für externe. Dies hat lange für Verwirrung gesorgt und wurde daher namensgebend zu Externe unbenannt. Dies war eine weitere Lektion dieses Mal im Thema Namensgebung.

Die geübte Datenbank Spezialist*in fragt sich jetzt wieso es eine UserChat Tabelle gibt, wenn alle Rechte über das Projekt laufen. Diese Frage habe ich mir auch ausgiebig überlegt. UserChat ist nicht zum Hinzufügen, sondern zum Entfernen. So kann ich mich selbst aus dem Chat entfernen oder den Chat Stumm stellen.

6.1.2. Backend

Das Backend wurde mit Python und dem Framework Django umgesetzt. Die Struktur ist daher im Voraus definiert und bietet keine grossen Freiheiten. Hier ist eine gekürzte Auflistung der wichtigsten Files. Es wurde ein Code First Ansatz gewählt. Also die Datenbank wird anhand der Modelle definiert.

File	Inhalt
models.py	Hier werden alle Datenbank Tabelle als Klasse definiert.
serializers.py	Hier wird definiert, wie das Model verarbeitet wird.
views.py	Hier wird unter anderem Zugriff eines Serialisiert geregelt.
urls.py	Hier wird die View als Endpoint bereitgestellt.

Tabelle 23

Folgende zwei Ergebnisse sind nennenswert aus diesem Teil:

- Bei der Entwicklung wurden fortlaufend alle Endpoints mit Postman getestet. Dieses Vorgehen hat sehr gut funktioniert. Dazu mehr im Testprotokoll.
- Erst während der Entwicklung ist mir Aufgefallen, dass ich gar kein Berechtigungssystem eingeplant habe. Wieder ein grosser Lernmoment.

6.1.3. Frontend / App - Pattern

Ohne klarem Vorgehen hier begonnen. Das hat sich gerächt. So musste mit einer besseren Methodik neu begonnen werden. Im Anhang Realisierung ist eine Nutzwert Analyse verschiedener Design Patterns. Diese hat ergeben, dass das MVVM-Pattern in diesem Projekt sich am besten eignet.

Directory	Inhalt
services	Hier sind alle Operationen zwischen Endpoints und Models
models	Alle Klassen basierend auf den Endpoint JSONs
widget	Hier wird Darstellung der Models festgelegt. (ViewModel)
views	Hier werden die ViewModels in Pages dargestellt.

Tabelle 24

6.1.4. Frontend / App - Fazit

Beim zweiten Anlauf wurde für jedes Model ein Service und ein Integration Unit-Test geschrieben. So konnte sichergestellt werden das alle Services funktionieren, bevor es weiter geht.

Folgende zwei Ergebnisse sind nennenswert aus diesem Teil:

- Unit Test haben sich als sehr praktisch erwiesen
- Methodisches Vorgehen hat sich bewährt und ergab ein Aha-Erlebnis

6.1.5. Kalender Synchronisation

Bei der Kalender Synchronisation gab es Probleme, welche aber nicht unlösbar waren. Google Calender synchronisiert nur alle 24h externe Kalender, dies kann nicht manuell angeworfen werden. Sie verweisen auf die API, welche sie bereitstellen. Anders als bei Apple Calender, hier lässt sich hier das Intervall auf 5min einstellen. Das Problem liess sich lösen mit einer App namens ICSx5. Hier kann man zu einem Freien Intervall die Synchronisation anstossen.

Damit ich dies auf dem Smartphone testen kann, reicht es nicht mehr die App lokal zu Testen. Spätestens jetzt erreicht die Container Landschaft seine aktuelle Form. Denn es gibt ein Problem mit 5G Routern, GCNAT und DDNS. Die Lösung ist detailliert in der Studie beschrieben. Die Lösung, welche genutzt wurde, ist der Cloudflare Tunnel und nginx Server.



Abbildung 10: Finale Container Architektur

6.1.6. Kanban

Der Rest der Realisierung ist reibungslos abgelaufen. Alle Aufgaben wie auch Notizen habe ich auf dem «Konkurrenten» Clickup festgehalten.

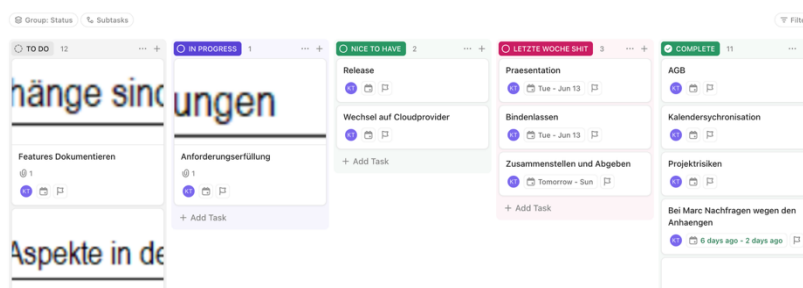


Abbildung 11: Kanban Board

6.2. Testprotokoll

Es gab mehrere Testdurchführungen. Diese wurden im Testprotokoll im Anhang festgehalten.

6.2.1. User- und Abnahmetest

Basierend auf Anforderungen der Applikation, wurde bereits in der Studie Abnahmekriterien definiert. Basierend darauf wurden Testfälle entwickelt. Diese Testfälle wurden nun einem User ohne Vorkenntnisse der Applikation auferlegt. Das Ziel war auch zu sehen, ob die Applikation selbsterklärend ist. Dies ist geglückt. Der User konnte alle Ergebnisse, ohne die Testschritte zu kennen ausführen.

Alle Muss Anforderungen wurden umgesetzt und konnten erfolgreich getestet werden. Es gab Abnahmekriterien, welche bis zum Ende nicht mehr gepasst haben. Die Templates sind aufgrund der Datenbank Architektur herausgeflogen.

6.2.2. Penetrationtests

Es wurden mehrere Penetration Test ausgeführt. Hier ist nur ein Auszug. Die Abdeckung ist auch noch zu klein. Es zeigt sich bisher, dass der Design by Security hat sich bewährt. Der Test auch noch gezeigt, beim einem Produktiven Start muss noch das Debug Modus beendet werden.

Test	Mängelklasse
Token - Geschützte Endpunkte ohne Token	0
Token - JWT Token Manipulation	0
Token - Zugriff auf fremde Daten	0
Premium - Admin-Funktionen für normale User	0
SQL Injection Tests	0
Grenzwerte und Datentypen	0
iCal Token Sicherheit	2
Error Message Information Leakage	4

Tabelle 25

7. Einführung

7.1. Wechsel auf Cloudprovider

Basierend auf dem Datenschutzkonzept und den Überlegungen aus der Studie wäre es für die Applikation am besten auf eine Managed-DB zu wechseln. Respektive auf einen DBaaS wie es bei Exoscale.com heisst. Ein Schweizer Cloud Provider mit Geo-Redundanz, Backups, Monitoring und eine Uptime von 99.9%. Das kann ich nicht für 1100 CHF bieten, dafür hätte ich länger als 13h pro Jahr. Der Wechsel muss einfach vor dem Release geschehen, um eine mühselige Migration zu umgehen. Sobald die Verbindung steht, können die Credentials im settings.py geändert werden und mit einem Command alle Migrationen wiederholt werden.

7.2. Betriebliche Umsetzung

Für den Produktionsbetrieb der Wechsel auf einen Managed Service (z. B. Exoscale) vorgesehen. Folgende Massnahmen ergaben sich aus dem Security by Design Konzept:

- 4h Backup der Datenbank (DBaaS bei Exoscale) mit 24-h-Retention.
- Disaster-Recovery-Strategie mit Backup-Restore-Test alle 6 Monate.
- 3-2-1-1-0 Strategie mit Scaleway Objectstorage und Air gapped Backup.
- Pentesting-as-a-Service jährlich durch externe Stelle (siehe Controlling).

Diese Massnahmen stellen sicher, dass im Falle eines Systemausfalls oder Datenverlusts der Betrieb innerhalb von Stunden wiederhergestellt werden kann.

7.3. Appstore Release

Das Endziel ist der Release in den Appstores. Dieses Ziel ist in den Hintergrund gerückt als ich mich zwei Wochen vor Abschluss der App von meiner Band trennte. Beim Versuch ein Account bei Goolge Console zu erstellen, muss man leider eine «richtige» Kreditkarte besitzen. Ich mit meiner Postfinance Mastercard konnte hier den Schritt noch nicht wagen. Den Aufwand ein neues Bankkonto zu erstellen, ist eine Aufgabe nach der Diplomarbeit. Der Applestore zwingt eine Feature List zu erstellen. Deshalb ist diese auch in der Realisierung gelandet.

7.4. Controlling

7.4.1. Nachkalkulation

Neu eingeflossen ist ein Jährliches Pentesting (Pentesting as a service) von einer Externen stelle. Das Marketingbudget ist bei 10% geblieben

Ist Kosten	CHF	%
Diplomarbeit	37080	34
Aufwand	49920	46
Marketingbudget	11000	10
Backup	43	0
Server	1570	1
Pentesting	10000	9
Total	109570	100

Tabelle 26: Nachkalkulation

Der grösste Unterschied, hier wurde mit 80 CHF anstelle von 120 CHF in der Stunde gerechnet. Die Diplomarbeit war noch 250 anstelle von 300. Backup und Pentesting war noch nicht dabei.

Soll Kosten	CHF	%
Total	57250	100

Tabelle 27

7.4.2. Break Even

Break Even ist immer noch im 5ten Jahr. Liegt am 10% Marketing hebel.

Jahr	Kosten (TCHF)	User	Umsatz (TCHF)	Saldo (TCHF)
1	109	438	31	-78
2	182	876	94	-87
3	254	1314	189	-65
4	327	1752	315	-11
5	399	2190	473	73
6	472	2628	662	190
7	544	3066	883	338

Tabelle 28

8. Schlussbetrachtung

Ich habe sehr viel gelernt. Ich froh habe ich das gemacht. Wenn ich nicht auf Marc gehört hätte, hätte ich noch gar nicht angefangen. Ich bin froh kann ich sagen, das nächste Mal bin ich klüger. Bei der Realisierung haben sich die «Fehler» als tolle Aha-Erlebnisse herausgestellt. Die Konzeptphase erscheint mir im Nachhinein am wichtigsten, war aber von der Aufteilung her etwas zu kurz. Die ist in der Planung etwas kurz gekommen. Datenschutzkonzept und Sicherheit-Konzept haben mich sehr bereichert. Testen hat mich auch positiv überrascht. Gefallen hat mir auch, eine Planung zu machen und am Ende die Unterschiede zu Vergleichen. Sehr hilfreich war es, dass ich für das Thema Band und Musik brenne.

Dokumentweise sind sicher 95% von mir selbst. Die 5% habe ich immer deklariert. Beim Code ist es etwa umgekehrt. Code besteht etwa 95% aus Claude. Jedoch habe ich meine 5% nicht deklariert. Abbildungen sind auch alle von mir selbst, Urheberrecht gehört dem entsprechend mir.

Ob das Projekt weiter geht, steht noch in den Sternen. Ich möchte das Finanzielle Risiko nur eingehen, wenn ich an die App glaube. Dank der Arbeit weiss ich, gibt es noch diverse Baustellen. Die Zukunft zeigt noch, ob ich Lust habe das Gebäude fertig zu stellen. Reizen würde mich es schon. Jedoch würde ich gerne noch Zeit mit Menschen verbringen, welche mich in dieser Zeit unterstützt haben.

Grossen Dank an meine Grossmutter und meine Mutter, welche zu meiner Tochter geschaut habe. Dank an meine Frau, welche mir Raum gelassen hat, mich daran erinnert und motiviert hat. Danke an Tim der mich mental unterstützt hat. Danke an Dominic mit dem ich mich austauschen konnte. Danke an meine Band welche mir diese Idea gegeben hat. Danke an meine Familie und Freunde, die noch mir Reden, auch wenn ich schon ewig nicht gemeldet habe. Danke an meine Experten und den Ausschuss, die das mitgemacht haben.

9. 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.

Bern, 09.06.2025

Kenan Tabinas

10. Anhang

Anhang A: Literaturverzeichnis, Quellenangaben

Anhang B: Diplomeingabe / Projektinitialisierungsauftrag

Anhang C: Projektauftrag

Anhang D: Terminplan

Anhang E1, E2 - En: Zusätzliche Konzeptsdokumente

Anhang F: Zusätzliche Informationen / Beilagen

Anhang G: Protokolle / Präsentationen

Anhang H: Logbuch

Anhang I: Abkürzungsverzeichnis, Glossar