|  |  |  |
| --- | --- | --- |
| Delegator | | |
| Kenan Tabinas | | |
|  | | |
| HFINF | Klasse 21b / Praktische Diplomarbeit 2025 | ID: 2558 |

# Management Summary

Im

Hier wird eine Zusammenfassung der ganzen Arbeit (und nicht für jeden Phasenbericht einzeln) erstellt. Dabei sollen auch Laien verstehen können um was es geht. Umfang im Maximum eine A4 Seite. Sie ist direkt nach dem Titelblatt einzufügen.

# Inhaltsverzeichnis

Contents

[1. Management Summary 2](#_Toc200236466)

[2. Inhaltsverzeichnis 3](#_Toc200236467)

[3. Initalisierung 6](#_Toc200236468)

[3.1. Ausgangslage 6](#_Toc200236469)

[3.2. Projektziele 7](#_Toc200236470)

[3.3. Projektorganisation 8](#_Toc200236471)

[3.4. Projektplan 9](#_Toc200236472)

[3.5. Lieferergebnisse 10](#_Toc200236473)

[3.6. Ressourcenplan 11](#_Toc200236474)

[3.7. Risiken 12](#_Toc200236475)

[3.8. Abgrenzung 13](#_Toc200236476)

[3.9. Studie 14](#_Toc200236477)

[3.9.1. Konkurenzanalyse 14](#_Toc200236478)

[3.9.2. Zielgruppenanalyse 15](#_Toc200236479)

[3.10. Technologie 16](#_Toc200236480)

[3.10.1. Varianten 16](#_Toc200236481)

[3.10.2. Entscheidung 17](#_Toc200236482)

[3.10.3. Bewertung 17](#_Toc200236483)

[3.10.4. Fazit 17](#_Toc200236484)

[3.11. Nicht Entscheidungen 18](#_Toc200236485)

[3.12. Service Stack 18](#_Toc200236486)

[3.12.1. Mögliche Varianten 18](#_Toc200236487)

[3.12.2. Break Even 19](#_Toc200236488)

[3.12.3. Fazit 19](#_Toc200236489)

[3.13. Wirtschaftlichkeit 20](#_Toc200236490)

[3.13.1. Konkurrenz 20](#_Toc200236491)

[3.13.2. Betriebskosten 20](#_Toc200236492)

[3.13.3. Instagram Ads 21](#_Toc200236493)

[3.13.4. Errechnete Neu-User 21](#_Toc200236494)

[3.13.5. Preismodell 21](#_Toc200236495)

[3.13.6. Breakeven 22](#_Toc200236496)

[3.13.7. Mögliches Risiko 22](#_Toc200236497)

[4. Konzept 23](#_Toc200236498)

[4.1. Designkonzept 23](#_Toc200236499)

[4.2. Berechtigungskonzept 23](#_Toc200236500)

[4.3. Testkonzept 23](#_Toc200236501)

[4.4. Datenschutzkonzept 23](#_Toc200236502)

[4.5. Sicherheitskonzept 23](#_Toc200236503)

[4.6. Datenbankkonzept 24](#_Toc200236504)

[4.6.1. Projekt 24](#_Toc200236505)

[4.6.2. User 24](#_Toc200236506)

[5. Realisierung 26](#_Toc200236507)

[5.1. Ausführung 26](#_Toc200236508)

[5.1.1. Backend Umgebung 26](#_Toc200236509)

[5.1.2. Datenbank 26](#_Toc200236510)

[5.1.3. Projekt Rechte 27](#_Toc200236511)

[5.2. Testprotokoll 28](#_Toc200236512)

[6. Einführung 29](#_Toc200236513)

[7. Schlussbetrachtung 29](#_Toc200236514)

[8. Authentizität 30](#_Toc200236515)

[9. Anhang 31](#_Toc200236516)

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

# Initalisierung

Vorwort, Einführung

## Ausgangslage

Das Leben in einer Band ist schwierig. Gerne möchten 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 auch ohne Organisation auch 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 Technik Uninteressierte 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 Zielgruppen gerecht lösen könnte.

## Projektziele

Folgende Ziele sollen bis Ende Projekt erreicht werden.

Beschreibung, was mit dem Projekt erreicht werden soll (Wirkung bzw. Nutzen).

Die Formulierung einer Zielvorgabe muss die folgenden Fragen beantworten können:

 Ist das Ziel genau und klar beschreiben?

 Ist das Ziel messbar?

 Welches Ergebnis soll erreicht werden?

 Ist das Ziel realistisch und bis Projektende erreichbar?

 Bis wann muss das Ergebnis vorliegen?

|  |  |  |
| --- | --- | --- |
| 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 der Zeit Rahmen 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 mein Kalender Synchronisieren |

Tabelle 1: Ziele

## Projektorganisation

Aufzeigen der Projektorganisation mit den entsprechenden Rollen.

 Aufzeigen wie im Projekt kommuniziert wird.

Ein Bild, das Text, Screenshot, Schrift, Electric Blue (Farbe) enthält.

KI-generierte Inhalte können fehlerhaft sein.

## Projektplan

Terminplan erstellen. Am klarsten und übersichtlichsten ist eine grafische Darstellung, wo Phasen, die wichtigsten Arbeitspakete und Meilensteine gezeigt werden. Am Ende des Projektes kann dem vorab erstellten SOLL-Terminplan noch der IST-Terminplan beigefügt werden (SOLL/IST - Vergleich ermöglichen). Der detaillierte Terminplan wird in den Anhang gelegt und im Diplombericht darauf verweisen.

Ein Bild, das Diagramm, Reihe, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.

## Lieferergebnisse

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

## Ressourcenplan

 Welche Mittel, Infrastruktur werden benötigt?

 Welcher Aufwand und welche internen/externen Kosten werden geschätzt?

 Was steht zur Verfügung?

 Mess-, Versuchs- und Testgeräte müssen genau protokolliert werden!

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

A white flower on a black background

AI-generated content may be incorrect.

Abbildung 1: Einfache Darstellung

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

Ein Bild, das Diagramm, Screenshot enthält.

KI-generierte Inhalte können fehlerhaft sein.

Abbildung 2: Server zu Internet

Es laufen fünf verschiedene Container die je einen Zweck erfüllen.

A screenshot of a computer

AI-generated content may be incorrect.

Abbildung 3: Darstellung der Container

## Risiken

A diagram of a number of colored squares

AI-generated content may be incorrect.Bei der folgenden Aufleistung gibt es ein Teil von den Risiken, um sich auf das Wesentliche zu begrenzen.

Abbildung 4: Risiko Matrix

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NR | Beschreibung | Eintritts-wahrschinlichkeit | 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 | Konkurenzverhalten | 1 | 3 | 6 |
| 6 | Datenschutzverletztung | 2 | 2 | 4 |

Tabelle 3: Risiko Analyse

## Abgrenzung

Für was ist dieses Projekt nicht gedacht?

 Bis wie weit soll dieses Projekt reichen?

 Was wird in diesem Projekt nicht bearbeitet?

 Für wen ist dieses Projekt nicht gedacht?

In der Folgenden Grafik wird die Grenze des Projektes visualisiert. Alle Elemente im Grünen Zylinder sind Teil vom Projekt. Der Focus 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 Risko einher welches, während dem Projekt nicht eingegangen wurde.

A diagram of a diagram of a cloud computing process

AI-generated content may be incorrect.

Abbildung 5: Abgrenzung

## Studie

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

### Konkurenzanalyse

Die Frage, die sich am Anfang stellt, steige ich bereits in einen Übersättigen 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 heraus.

|  |  |  |
| --- | --- | --- |
| 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 4

Asana und Clickup erschienen als Dorn im Auge. Beim Versuch die App zu Testen und auf eine Feature zu überprüfen. Nämlich die Kalender Synchronisation. Dabei ging für das Projekt in kleine Richtung.

Clickup ist komplett überladen. Die Einstellungsmöglichkeiten scheinen schier unendlich. Das mag zuerst wie ein Vorteil klingen, jedoch ist auch ein Nachteil. Daraus haben sich folgende erkenntisse ergebn:

* 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.

### Zielgruppenanalyse

Bei der Vorstellung des Projektes war die Primäre Zielgruppe Musiker. Marc Abby und Ralph Maurer sprachen davon, dass dieses Produkt einen Nerv trifte. 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 5

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 6

## 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.

Folgende Punkte wurden beachtet:

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

### Varianten

Folgende Technologien Varianten standen zur Auswahl:

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

A diagram of a computer system

AI-generated content may be incorrect.

Abbildung 6: Technologievarianten

### Entscheidung

Der gesammte Variantenscheid 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 von Goolge Playstore und Apple Appstore. Zusammenfassend kann man sagen wegweisend waren die vorgängigen Analysen. Diese haben der Userexperience einen Hohen Stellen wert zu gesprochen.

### 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 7: B2 Bewertung

### 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 Unterhalten.

Eine Frage welche noch nicht geklärt ist. Welchen Ansatz der Zielführende ist. Bei genauerem Hinsehen ist die WebViewApp und Cross-Plattform nahe beieinander. Für die Entwicklung von WebView Apps bietet sich die Verwendung von Cross-Plattform-Framework an.

Die User Experience liefert den Finale Entscheidung. Die User Experience auf einem 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.

## 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 8: 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 9: Deployment-Architekturen

## Service Stack

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

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

### Mögliche Varianten

|  |  |
| --- | --- |
| Variante | Bezeichnung |
| V1 | On-Premise |
| V2 | VPS |
| V3 | CaaS |

Tabelle 10: Service Stack

### Break Even

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

Wenn ich von 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.

### Fazit

Bis zum Ende war ich davon überzeugt, dass nichts günstiger ist als On-Premise. Da eine 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 klar die Entscheidung ist auf einen Managed Service gefallen.

## Wirtschaftlichkeit

### 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 11

### Betriebskosten

In der Studie ist detailliert aufgelistet und beschrieben wies es zu folgenden Zahlen gekommen ist. Relevant für das Verständnis, sind die 10% für das Marketing, dieses 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 12

### 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 |

### 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 13: Instagram CPA bei 11%

### 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 auf einem 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 14: Preimodell

### 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 15: Breakeven Studie

### Mögliches Risiko

Nehmen wir ich rechne meine Zeit nicht. Ich rechne nur Kosten, welche ich an für Werbung und Server zahle. Im aller Worstcase bezahle ich im ersten Jahr mal 7000 CHF ohne etwas zurück bekommen zu haben.

Abbildung 7: Breakeven Kosten

# Konzept

## Designkonzept

Bei Design Prozess ist in 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.

## Sicherheitskonzept / Berechtigungskonzept

## Testkonzept

## Datenschutzkonzept

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

## Datenbankkonzept

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

### Projekt

Der Kern Datenbank ist das Projekt. Alle Detail Einträge sind modular und besitzen das 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 inform von meistens Fotos. |

Tabelle 16

### 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 17

Konzeptionelle Aussagen zur Lösung

Abbildung 8: Datenbank Modell

• Erklärung zur Lösungsarchitektur   
• Erklärung der technischen Umsetzung   
• Erklärungen wie die Tests durchgeführt werden   
• Erklärungen wie die Security und Datenschutz gewährleistet wird   
• Erklärungen was für den Betrieb der Lösung beachtet werden muss

Design der Lösung   
• Grob- und Detaildesign, Prozesse, Abläufe etc. erstellen. Alle Zusammenhänge müssen nachvollziehbar und transparent sein!   
  
Wenn mehrere Konzepte als einzelne Dokumente erstellt wurden, kann im Diplombericht aufgezeigt werden welche Konzepte erstellt wurden und was deren Inhalt ist. Die jeweiligen Konzepte können in den Anhang gelegt und im Diplombericht darauf verwiesen werden

# Realisierung

## 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, drei grossen Aha-Momente geführt haben.

### Projekt Rechte

Was am Anfang zur Verwirrung beigetragen hat, ist die Tabelle UserProject. Man könnte meinen diese listet alle Projekt Teilnehmer auf. Jedoch wurde sich während des Designs Phase gegen eine Kompliziertes Rollen- und Berechtigung-System entschieden. Lustigerweise ist auch eines daraus geworden, jedoch nicht für den User.

Wenn alle jeder Band-Mitgleider zu jedem Projekt hinzugefügt werden müssen. Füllt es die Tabelle UserProjekt immer gleichen Einträgen. Diesen Overhead wurde durch ein anderes System ersetzt.

Alle User werden via Tabelle User Organisation der Organisation hinzugefügt. Sie können nach 5 Verschiedenen Rollen hinzugefügt werden. Sobald sie Teil der Organisation sind, erhalten sie Automatische Zugrife.

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

Tabelle 18: 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 dies Mal in Thema Namensgebung.

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

### 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 Serialisierst geregelt. |
| urls.py | Hier wird die View als Endpoint bereitgestellt. |

Tabelle 19

Folgende zwei Ergebnisse sind nennenswert aus diesem Teil:

* Bei der Entwicklung wurde 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 Lern Moment.

### Frontend / App - Pattern

Ohne klarem Vorgehen wurde 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 20

### 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

### 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 Manuel angeworfen werden. Sie verweisen auf die API, welche sie bereitstellen. Anders als bei Apple Calender, hier lässt sich 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 dies auf den 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.

A screenshot of a computer

AI-generated content may be incorrect.

Abbildung 9: Finale Container Architektur

### Kanban

A screenshot of a computer

AI-generated content may be incorrect.Der Rest der Realisierung ist reibungslos abgelaufen. Alle Aufgaben wie auch Notizen habe ich auf dem «Konkurrenten» Clickup festgehalten.

Abbildung 10: Kanban Board

## Testprotokoll

Ausführen

• Prototyp erstellen und erklären   
• Erstellen der benötigten Marketing- und Finanzdokumente wie Factsheet, SLA, Businesscase mit ROI Berechnung, Break Even Analysen etc.

Tests

• Planung: Was wird getestet? Welche Tests sollen durchgeführt werden (Test Cases erstellen)? Welches Resultat wird erwartet?   
• Durchführung: Testprotokoll erstellen   
• Protokollierung (wie wurde getestet): Die Testanordnung muss ersichtlich sein. Das Test-Equipment (Geräte, div. Material) muss erfasst und aufgelistet werden.   
• Testergebnisse müssen immer nachvollziehbar und beweisbar sein!   
• Auswertung, Testbericht

Es kann auch nur der Inhalt vom Testkonzept und Testprotokoll aufgezeigt werden. Dann werden diese Dokumente in den Anhang gelegt und im Diplombericht darauf verwiesen.

# Einführung

## Wechsel auf Cloudprovider

Basierend auf Datenschutzkonzept und den Überlegungen aus der Studie wäre es für den Applikation am besten auf eine Managed DB zu wechseln. Respektive auf einen DBaaS wie es bei Exoscale.com heisst. Ein Deutscher 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.

Beim Service wäre es auch möglich, auf dem Raspi 5 wieder als Server zu verwenden. Durch die Cloudflare Architektur ist der Service vor DDoS geschützt. Sämtliche Verbindungen sind immer durch Cloudflare Encryptet. Durch den Zero Trust Tunnel blieben alle Ports geschlossen. Monitoring wird von Cloudflare bereitgestellt. Folgende Einstellung müssen trotzdem vorgenommen werden:

* Server in separates VLAN
* Auto Updates einschalten Server
* Least Privilege auf dem Container
* Separates Monitoring einrichten
* Penetration Tests

## 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.

## Controlling

### Nachkalkulation

Neu eingelossen ist ein Jährliches Pentesting (Pentesting as a service) von einer Externen stelle.

| Position | CHF | % |
| --- | --- | --- |
| Diplomarbeit | 24000 | 29 |
| Aufwand | 33280 | 59 |
| Marketingbudget | 6000 | 10 |
| Server | 1 | 2 |
| PTaaS | 10000 |  |
| **Total** | **64850** | **100** |

Tabelle 21: Nachkalkulation

• Darlegung der tatsächlichen Kosten / Renditen mit einer Nachkalkulation   
• Diskrepanz zur ursprünglichen Budgetplanung aus der Studie aufzeigen und begründen   
• Abschliessende wirtschaftliche Betrachtung der Arbeit

# Schlussbetrachtung

• Schlusskommentar zum Ergebnis der gesamten Arbeit   
• Darstellen welchen persönlichen Beitrag in die Lösung eingeflossen ist   
• Wie geht es weiter mit dem Projekt   
• Persönliche Betrachtung   
• Dank   
• Urheberrecht

# Authentizität

Der folgende Text muss unbedingt im Diplombericht enthalten sein:

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

• Ort, Datum, Unterschriften der Diplomanden! (nicht der Experten)

# Anhang

Anhang A: Literaturverzeichnis, Quellenangaben, Kontakte

Anhang B: Diplomeingabe / Projektinitialisierungsauftrag

Anhang C: Projektauftrag Anhang D: Terminplan Anhang E1, E2 - En: Zusätzliche Konzeptdokumente

Anhang F: Zusätzliche Informationen / Beilagen

Anhang G: Protokolle / Präsentationen

Anhang H: Logbuch

Anhang I: Abkürzungsverzeichnis, Glossar