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

# Management Summary

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.

Das Unternehmen Delegator wurde ins Leben gerufen um ein bekanntes Problem von kleinen Unternehmen, Vereinen und Familien zu lösen.

Der aktuelle Markt besteht aus Kalender welche das Erfassen von Aufgaben nicht möglich ist. Kanban Boards Erfassung von Ressourcen. Projektmanagement Tool welche ein Vermögen kosten.

**Die Zielgruppe:**  
Wem ich helfen möchte, sind Musiker. Musiker welche nicht gerade bekannt dafür sind die Strukturierten zu sein. Die Zeiten von Sex Drugs and Rock’n’Roll sind etwas in die Jahre gekommen. In einem Markt, wo jeder seine eigene Musik ohne Label Boss Verbreiten kann, braucht es trotzdem einen Manager. Am liebsten einer, der nicht 15% aller Einnahmen nimmt.

**Das Geschäftsmodell:**  
Verkaufen ist Schnee von gestern. Abo Modelle sind die Zukunft. Doch alles hinter einer Paywall verstecken, lockt keine neuen Interessenten an. Daniel Priestley empfielt das 10/90 Buisnessmodell. Mehr dazu wie ich es mir vorstelle im entsprechenden abschnitt.

**Das Kurzfriste Ziel ist es:**  
Aufgaben werden in einem Kanban Board erfasst und automatisch mit einem Kanban Board Synchronisiert. Zu dem sollte es möglich sein, mehrere User auf die Applikation einzuladen und ihre Ressourcen einzutragen. In ferner Zukunft möchte ich die Möglichkeit bieten, Dateien hochzuladen und zu Versionieren und eine Art Kommentar Funktion anzubieten in Form eines Chats. Was genau Teil der Arbeit ist, ist sichtbar im entsprechenden Bereich.

# Inhaltsverzeichnis

Contents

[1. Management Summary 2](#_Toc191147353)

[2. Inhaltsverzeichnis 3](#_Toc191147354)

[3. Initalisierung 4](#_Toc191147355)

[a. Ausgangslage 4](#_Toc191147356)

[b. Projektziele 5](#_Toc191147357)

[c. Projektorganisation 6](#_Toc191147358)

[d. Projektplan A 7](#_Toc191147359)

[e. Projektplan B 8](#_Toc191147360)

[f. Lieferergebnisse 8](#_Toc191147361)

[g. Risiken 8](#_Toc191147362)

[h. Resourcenplan 8](#_Toc191147363)

[i. Abgrenzung 9](#_Toc191147364)

[j. Studie 9](#_Toc191147365)

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

# Offene Fragen

ChatBot -Sevice Hotline. Wenn ich eine Hotline mache muss ich zwar immer ereichbar sein und habe auch die Kosten. Ist somit kosten aufwendiger beim Skalieren ist aber nahbarer für Kunden. Das ist vor allem ein Positionierungs Frage.

# Initalisierung

Vorwort, Einführung

## Ausgangslage

 Welches ist die momentane Situation?

 Weshalb soll das Projekt realisiert werden?

(Keine Beschreibung der Lösung sondern den Nutzen aufzeigen!)

 Für wen (Zielpublikum) ist dieses Projekt gedacht, was bezweckt der daraus

resultierende Bericht?

Das Organisieren von Aufgaben ist unerlässlich, sobald mehrere Personen involviert sind.

Das Unternehmen Delegator hat sich zum Ziel gesetzt, eine innovative Lösung zu entwi-ckeln, die die Zusammenarbeit für Einzelpersonen, Familien, Vereine und Unternehmen effi-zienter und einfacher gestaltet. Der Fokus liegt darauf, bestehende Lücken zwischen belieb-ten Kollaborationstools wie Microsoft Teams, ClickUp und Trello zu schliessen. Diese Werk-zeuge bieten zwar viele nützliche Funktionen, stossen jedoch an ihre Grenzen, wenn es da-rum geht, Kalender und Kanban-Boards zu synchronisieren oder Ressourcen einzutragen.

Die aktuelle Marktsituation zeigt, dass Alternativen wie Asana und Jira zwar leistungsstark sind, jedoch für viele Zielgruppen zu kostspielig. OpenProject wäre zwar kostenlos wirkt aber etwas altmodisch und wenig benutzerfreundlich.

## Projektziele

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 |

## 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 A

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.

## Projektplan B

|  |  |  |
| --- | --- | --- |
| **ID** | **Ziel** | **Bis wann?** |
| Z001 | Kick Off | 14.02 |
| Z002 | Initialisierung |  |
| Z003 | Zwischen-Meeting | 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 |

## 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 |

## Risiken

• Was für Risiken bestehen im Projekt?

• Was wird ausgeführt, damit diese Risiken nicht eintreffen?

• Was sind die Massnahmen die im Projekt getroffen werden wenn die Risiken

eintreffen?

## Resourcenplan

 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!

## 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?

## Studie

### Buisness Model Canvas

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Key Partnerships** | **Key Activites** | **Value Proposition**  Gemeinsame Kalender, wiederkehrende Aufgaben und erinnerungen.  Einkaufslisten, Social Media Planing, Meal Planner. | | **Customer Relationships** | **Customer Segments**  Bands, Familien, Vereine |
| **Key Resources** | **Channel** |
| **Cost Structure** | | | **Revenue Streams** | | |

### Markanalyse

Ich habe mir andere Player angeschaut und welche Pricing Modelle sie nutzen

|  |  |  |
| --- | --- | --- |
| Programm | Pricing | Unterschiede |
| Trello | Free Reduziert  5/ User / Jahr | Aufgaben sind nicht zuteil bar auf einzelne Personen in der Grund Version |
| Asana | Bis zu 15 Personen Frei  11 / User / Jahr |  |
| Google Calender | Kostenlos | Es ist möglich Aufgaben zu Terminieren. Genau gleich wie in Teams tauchen die Aufgaben nicht im Kalender auf. |
| ClickUp | Begrenzt Kostenlos  7 / User/ Jahr | Clickup kann sehr viel von dem ich auch einbauen möchte. Meine Applikation muss sicher Kosten güstiger sein als. Clickup.  Zuerst hat es mir etwas sorgen gemacht. Weil ich angst habe mein ganzes Konzept geht nicht auf, wenn ClickUp alles besser kann. Als ich probiert habe es mit einem Google Calender zu Synchronisieren, was funktionieren sollte, wurde ich etwas überfahren von der Funktionsumfang von ClickUp.  Da habe ich begriffen, dass muss der grösste Unterschied sein. Die Usability sein.  Das mag für IT-Gurus genau das richtige sein. Und für mich wichtig bei der Positionierung.  Es muss einfacher sein und IT Schtzuzige Bands, Familien und Unternehmen sind das Ziel.  Daher habe ich auch realisiert dass die Software zwingen Narren sicher sein muss und Qualität entscheident ist, wenn die Kunden die Software nicht verstehen oder zu lange warten müssen, verliere ich das Vertrauen.  Das hat auch gerade in die Frage Service Hotline eingespielt. Es muss Tutorials geben wie die App bedient werden kann. Ein ChatBot der häufige Fragen beantwortet und eine Hotline die gerne Fragen beantwortet und hilft. |

Es braucht ein Feedback möglichkeit und ein Bug-Reporting. Mit z.B. einem Goodie. Wenn ein Bug basierend auf ihrem Ticket gelöst wird, gibt es erlass auf eine Monats beitrag.

Während dem habe ich mir auch gedanken gemacht, dass es Strategisch sein könnte eventuell den Focus neu zu Orientieren auf Familien.

Familien gibt es mehr. Laut folgendem Wikipedia Artikel gibt es   
<https://de.wikipedia.org/wiki/Musik_in_Deutschland>

Gibt es ca. 500'000 Bands in Deutschland. Es gibt jedoch 12 Millionen Familien.

Ichhabe mir noch angeschaut welche Tools es für Familien gibt.

|  |  |  |
| --- | --- | --- |
| Cozi Familly Organizer | 5 / Monat | Einkaufslisten |
| OurHome | Kostenlos  Keine Werbung? | Belohnungsystem  Haushaltsaufgaben |
|  |  |  |
|  |  |  |

OurHome hat mich etwas überrascht, bin mir nicht sicher wie das Rentabel ist. Erst war ich etwas eingeschüchtert von OurHome. Ich suchte nach Reviews. Bis mir aufgefallen ist, es existiert nicht im AppleStore. Daraus habe ich gelernt mache ich etwas ähnliches muss es dies abdecken.

Bei der weiteren Recherche ist mir aufgefallen das ist ein Spanennder Markt der noch etwas Luft nach oben hat.

Was jedoch klar ist es gibt zwar 24mal mehr Familien als Bands. Es gibt dafür weniger Konkurenz.

Es gibt FamiliyWall, TimeTree, Cozi Family, FamiliyNest, Maple: Family Organizer, MyFamiliz family Calender, Octave Family Organizer, Fam Cal: Family Shared Calender, Share(d), the Faimily App. Der Markt ist gesätigt. Eventuell nicht übersättigt, aber gesättigt.

Hingegen wenn man nach einem Organisations Tools für Bands sucht:   
Gibt es diverse Aufnahem Tools, ein App welche dir sagt welche Bands in der Stadt sind, Tools für Tonleitern, Tools um auf dem Smartphone GuitarHero ähnliche Spiele zu spielen, Sport Apps mit Bändern, Setlist Manager und vieles mehr fast nichts zu dem ich machen möchte.

Zwei Apps habe ich mir trodem zu meiner Konkurenz analyse angeschaut:

|  |  |  |
| --- | --- | --- |
| BandHelper | Basic:  5/month  Plus:  7 / Month  Pro:  9 / Month | Es bietet an deine Stücke mit Noten zu darstellen, Calender, Kontakte, Finanzen Checkliste |
| Band Mule | Pricing nicht ersichtlich.  Es ist Free die ersten zwei Monate. Nachher nicht klar | Übungs Kalender, Chat, Abstimmungen für Übungen. Setliste online. |

Daraus schliesse ich Setlisten und übungen sind Wichtig. Das Abstimmungstool gefällt mir auch. Wenn ich über die Setlisten nachdenke, sollten sie auch gerade Versentbar sein, quasi ein PDF. Vielleicht wäre eine E-Mail funktion auch Praktisch, wenn man sich gerade mit SMTP oder so anemlden könnte. Das ich wirklich alles aus der App machen könnte.

|  |  |  |  |
| --- | --- | --- | --- |
| Nr | Feature | Muss / Kann | Begründung |
| F001 | Setlist | Muss | Die Konkurenz kann es auch |
| F002 | Chat | Kann | Ich glaube Whatsapp und co reicht. |
| F003 | Finanzen | Kann | Wäre gut. Nicht Dringend.l |
|  |  |  |  |
|  |  |  |  |

### Technologien

Es gibt 4 Varianten wenn wir diese APP aus dem Boden stampfen.

Entweder verzichte ich genzlich auf eine Mobile App und erstelle nur eine

|  |  |  |  |
| --- | --- | --- | --- |
| Name |  |  |  |
| Web Only |  |  |  |
| Advanced Web App |  |  |  |
| Native Development |  |  |  |
|  |  |  |  |

#### Webonly

Diese Varainte ist basiert auf einem Webauftritt. Die möglichkeit die Applikation über ein Smartphone App zu bedienen wird komplett wegelassen.

##### Vorteil

* Codebasis: Die Menge an Code wird massiv reduziert, da nur eine Plattform entwickelt und gewartet werden muss.
* Produktionszeit: Die Entwicklungszeit verringert sich erheblich, da keine separaten Apps für verschiedene Betriebssysteme programmiert werden müssen.
* Wartung & Updates: Änderungen können zentral durchgeführt und sofort für alle Nutzer verfügbar gemacht werden, ohne dass App-Updates verteilt werden müssen.
* Plattformunabhängigkeit: Die Anwendung ist von jedem Gerät mit Internetzugang und einem Browser erreichbar. Die höchste Unabhängigkeit von allen Varanten.
* Kostenersparnis: Weniger Entwicklungsaufwand bedeutet geringere Kosten für Programmierung, Wartung und Support.

##### Nachteil

* UX: Smartphone-Nutzer sind eher daran gewöhnt, native Apps zu verwenden. Eine Weblösung kann als weniger intuitiv empfunden werden.
* Performance: Eine Webapplikation ist in der Regel langsamer und als eine native App..
* Zugänglichkeit: Ohne eine Präsenz in den App Stores ist es schwieriger, neue Nutzer zu gewinnen, da viele Nutzer direkt im Store nach Lösungen suchen.
* Eingeschränkter Zugriff auf Gerätefunktionen: Native Apps haben direkten Zugriff auf Funktionen Push-Benachrichtigungen und Offline-Speicherung.

#### Webview App

Diese Variante basiert auf einer **WebView-App**, also einer nativen Anwendung, die im Grunde nur eine Webbrowser-Komponente enthält und eine Webanwendung darstellt.

**Vorteile:**

* Entwicklung: Da die eigentliche Anwendung eine Webseite ist, muss nur eine einfache native Hülle entwickelt werden.
* Geringere Kosten: Die WebView-App erfordert deutlich weniger Entwicklungsaufwand als eine vollständige native App.
* Plattformübergreifend: Die App läuft auf verschiedenen Betriebssystemen und Geräten, verglichen zu Nativen Apps.
* Automatische Updates: Änderungen an der Webanwendung werden direkt übernommen. Die App muss nicht jedes Mal im AppStore geupdatet werden
* Features: Mehr Features verglichen zu einer Web-Only Anwendung
* AppStore: Die Anwendung ist verfügbar im AppStore und macht sie zugänglicher als nur eine Webseite.

**Nachteile:**

* Entwicklung: Verlglichen zur WebOnly anwendung muss, mehr entwickelt werden.
* Schlechte UX im Vergleich zu nativen Apps: Nutzer erwarten eine native Benutzerführung und Performance, die eine WebView-App oft nicht bieten kann.
* Abhängigkeit vom Internet: Da die App eine Webseite lädt, ist eine permanente Internetverbindung erforderlich. Offline-Funktionalitäten sind nur mit zusätzlichem Aufwand realisierbar.
* Performance-Probleme: WebView-Apps sind oft langsamer als native Apps, da sie auf die Rendering-Engine des Webbrowsers angewiesen sind.
* App Store-Richtlinien: Apple und Google lehnen manchmal einfache WebView-Apps ab, wenn sie nicht genügend native Funktionen bieten.

#### Native App

Bei dieser Variante verwendet man 3 Verscheidene Codesbases. Ein WebFramework, Swift für iOS, Kotlin für Android.

##### Vorteile:

* Performance: Die Indviduelle Entwicklung macht die Performance 5-10 mal schneller als z.B React Native.[[1]](#footnote-1)
* UX: Native Apps sind intuitiv und schnell. Das Design passt genau zum jeweiligen Betriebssystem.
* Features: Zugriff auf alle Kamera, GPS, Bluetooth, Push-Benachrichtigungen etc.
* Offline-Funktionalität: Die App ist komplett Offline verwendet bar.
* App-Store: Native Apps entsprechen auch den Store-Richtlinien von Apple und Google

##### Nachteile:

* Massive Entwicklungskosten: Drei verschiedene Code-Bases ist ein Alptraum.
* Massive Entwicklungszeit: Die Erstellung und Wartung von drei separaten Apps ist astronomisch.

#### Cross Plattform

Diese Variante basiert auf einer Cross-Plattform-App, die mit einer einzigen Codebasis für mehrere Betriebssysteme (Android, iOS, Windows, macOS, Linux) entwickelt wird.

##### Vorteile:

* Entwicklungsaufwand: Eine einzige Codebasis spart Zeit und Geld.
* Plattformübergreifend: Die App läuft auf verschiedenen Betriebssystemen und Geräten, verglichen zu Nativen Apps.
* Wartung und Updates: Änderungen und Bugfixes müssen nur einmal gemacht werden. Der Wartungsaufwand reduziert sich massiv.
* Performance: Moderne Frameworks bieten vergleichbare Performance verglichen zu Nativen Apps.
* Features: Mehr Features verglichen zu einer Web-Only Anwendung
* AppStore: Die Anwendung ist verfügbar im AppStore und macht sie zugänglicher als nur eine Webseite.

##### Nachteile:

* Performance: Die Performance ist zwar gut. Jedoch nicht so gut wie bei Nativen Apps.
* Individuelle Anpassungen: Da jede Plattform hat anders ist, müssen im Zweifel Anpassungen gemacht werden.
* Abhängigkeit: Die App ist von der Weiterentwicklung der gewählten Cross-Plattform-Frameworks abhängig.
* Speicherverbrauch: Manche Frameworks benötigen zusätzliche Laufzeitbibliotheken, was die App-Grösse vergrößern kann.

#### Kriterien

##### Anforderungen

Bassiernd auf dieser Aufstellug der Option haben sich folgende Anforderungen ergeben:

* Entwicklung: Kosten beim start der Entwicklung
* Updates: Kosten bei Wartung und weiter Entwicklung
* Appstore: Verfügbarkeit im Appstore

##### Keine Anforderungen

Folgende Punke haben es nicht Rein Geschaft:

* Performance: Die Performance ist vernachlässigbar
* Native Features : PushUp Benachrichtigungen sind ein nice to have.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Web-Only | WebView | Nativen | Crossplattform |
| Entwicklung | 3 | 3 | 1 | 2 |
| Updates | 3 | 3 | 1 | 3 |
| Appstore | 1 | 2 | 3 | 3 |
| Total | 7 | 7 | 5 | 7 |

Die Native Entwicklung ist aus dem Fenster geflogen.

#### Webfirst Ansatz

Es gibt in jedem Fall eine Mobile App und in jedem Fall eine Webseite. In diesem Fall wird die Mobile App Zweitranging.

##### Vorteil

* Web UX: Das Benutzererlebnis auf der Webseite ist sicher besser als auf Mobile-First.

##### Nachteil

#### Mobilefirst Ansatz

Es gibt in jedem Fall eine Mobile App und in jedem Fall eine Webseite. In diesem Fall wird die Weboberfläche Zweitranging.

##### Vorteil

* Mobile UX: Das Benutzererlebnis auf
* Appstore: Eine Vollständige App entspricht auch den Richtlinien der Stores.

##### Nachteil

* Webansicht: Die Webansicht von Crossplattform Apps leided etwas, weil der dargestellte Inhalt ein Canvas ist und nicht übliches HTML.
* Web-Entwicklung: Wenn ich gewisse Funktionalitäten umsetzen möchte welche mir die Crossplattform App erlaubt, muss ich zusätzlichen Code schrieben.
* Erfahrung: Ich habe noch nicht so viel Erfahrung mit Crossplattform Frameworks.



Hier werden Grundsatzentscheide gefällt, es werden keine Details beschrieben!

 Informationsbeschaffung zur aktuellen Situation und Lösungsansätze aufzeigen

 Liste mit den funktionalen oder nicht funktionalen Anforderungen mit Sicht Benutzer, Systemadministration und Prozesse (Pflichtenheft) oder User Stories

 Beschreibung der einzelnen Lösungsvarianten mit grafischer Darstellung

 Analyse welche Lösungs-Varianten möglich sind

 Evaluation der möglichen Varianten mit einer Nutzwertanalyse

 Entscheid mit welcher Variante das Konzept realisiert wird

 Wirtschaftlichkeit über den ganzen LifeCycle der jeweiligen Varianten

(Total Cost of Ownership TCO)

Ausführliche Informationen wie Datenblätter, Anforderungsüberprüfungen etc.

werden in den Anhang gelegt und im Diplombericht darauf verwiesen.

1. <https://medium.com/swlh/flutter-vs-native-vs-react-native-examining-performance-31338f081980> [↑](#footnote-ref-1)