



# FOM Hochschule für Ökonomie und Management

Hochschulzentrum München

## **Seminararbeit**

Im Rahmen des Moduls

Arbeitsmethoden und Softwareunterstützung

Über das Thema

## **Mein Thema**

von

**Vorname nachname**

Gutachter: Dr. Herbert Bauer  
Matrikelnummer: 123456  
Abgabedatum: 09.01.2022

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>II</b>
<b>Abbildungsverzeichnis</b>	<b>III</b>
<b>Tabellenverzeichnis</b>	<b>IV</b>
<b>Abkürzungsverzeichnis</b>	<b>V</b>
<b>1 Vorwort</b>	<b>1</b>
1.1 Teambildung	1
1.2 Projektinitiierung	1
1.3 Projektskizze	1
<b>2 Ziele</b>	<b>2</b>
2.1 Muss-Ziele	2
2.2 Soll-Ziele	2
2.3 Kann-Ziele	2
2.4 Nicht-Ziele	2
<b>3 Stakeholderanalyse</b>	<b>3</b>
3.1 Vorgehen Stakeholderanalyse	3
3.2 Grafische Darstellung Stakeholderanalyse	3
<b>4 Risikoanalyse</b>	<b>5</b>
4.1 Vorgehen Risikoanalyse	5
4.2 Grafische Darstellung Risikobewertungsmatrix	5
<b>5 Implementierung des Chat-Screens</b>	<b>7</b>
5.1 Anforderungen an den Chat-Screen	7
5.2 Erstellung einer Flutter Widget-Seite	7
5.3 Integration des Stream-Chat-Flutter Pakets	8
5.4 Benutzerauthentifizierung	8
5.5 Implementierung der Chat-Widgets	8

# Abbildungsverzeichnis

Bild 1 : Stakeholderanalyse .....	4
Bild 1 : Risikomatrix .....	6

# Tabellenverzeichnis

# Abkürzungsverzeichnis

**SAE** *Society of Automotive Engineers*

**km/h** Kilometer pro Stunde

**t** Tonnen

**NO** Stickoxide

**HC** unverbrannte Kohlenwasserstoffe

**H<sub>2</sub>O** Wasser

**CO<sub>2</sub>** Kohlenstoffdioxid

**CO** Kohlenmonoxid

# **1 Vorwort**

## **1.1 Teambildung**

## **1.2 Projektinitiierung**

## **1.3 Projektskizze**

## **2 Ziele**

### **2.1 Muss-Ziele**

### **2.2 Soll-Ziele**

### **2.3 Kann-Ziele**

### **2.4 Nicht-Ziele**

## 3 Stakeholderanalyse

Für die STH App sind Stakeholder aus verschiedenen Bereichen vorhanden. Die Applikation erstrebt einen großen Einfluss auf die Sportindustrie und das speziell auf den Prozess des Anwerbens von neuen Fußballspielern durch Vereine in unterschiedlichen Größen.

Deshalb ist eine Stakeholderanalyse besonders wichtig, um die verschiedenen Gruppen an Interessenten zu identifizieren und zufriedenstellen zu können. Für die Analyse werden folgende Gruppen an Stakeholdern betrachtet: Interne-, Externe-, Community- und Technische Stakeholder.

### 3.1 Vorgehen Stakeholderanalyse

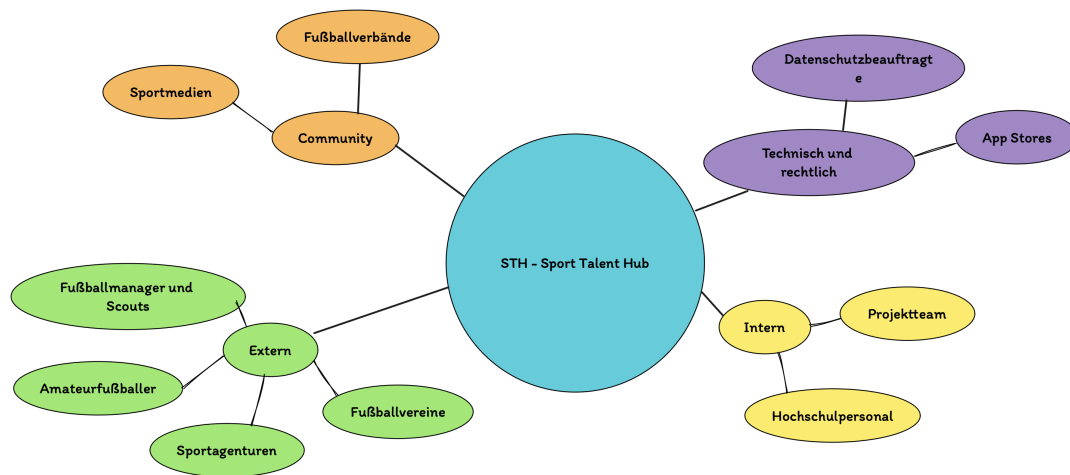
Die Stakeholderanalyse wurde mit einem konkreten Vorgehen erstellt. Zuerst wurde erneut die Größe und Auswirkung der Applikation betrachtet. Dabei ist es auch wichtig, die Zielsysteme nicht außer Acht zu lassen, um konkrete Benutzergruppen definieren zu können.

### 3.2 Grafische Darstellung Stakeholderanalyse

In der folgenden Abbildung sind die Stakeholdergruppen und deren Abhängigkeiten aufgezeichnet.



Bild 1: Stakeholderanalyse



Quelle: Quelle Stakeholderanalyse

## 4 Risikoanalyse

Die Risikoanalyse dient dazu kritische Einflüsse innerhalb und außerhalb des Projektes zu identifizieren. Für die STH App gilt vergleichbar zu allen anderen Projekten, dass der Projekterfolg nur mit begleitenden Risiken ermöglicht werden kann. Für die Klassifizierung und Einschätzungen der Risiken dient eine Risikomatrix. Hierbei wird aufgezeigt welche Risiken mit welcher Eintrittswahrscheinlichkeit und zugehöriger Auswirkung eintreten können. Anhand dessen kann bewertet werden, welche Risiken besonders laufend beobachtet werden müssen und ob es Risiken gibt, die den Projekterfolg maßgeblich gefährden.

### 4.1 Vorgehen Risikoanalyse

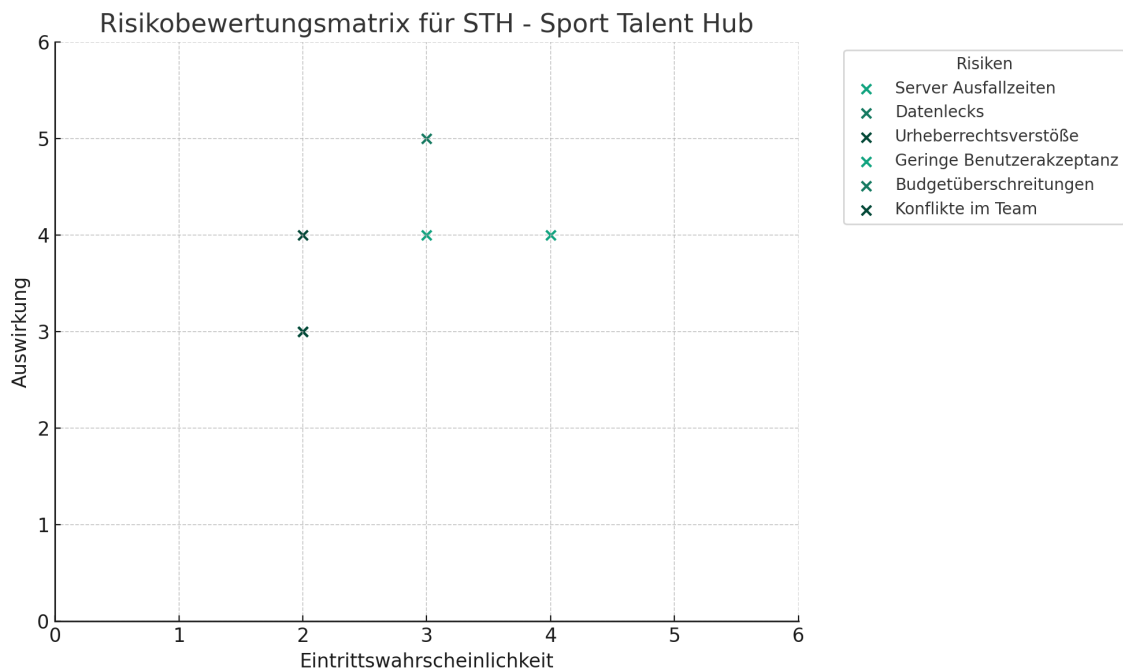
Um eine Risikobewertungsmatrix erstellen und veranschaulichen zu können benötigt es einige Schritte. Zunächst müssen die Risiken erkannt werden, die im Projekt auftreten können. Dabei werden sowohl interne als auch externe Faktoren beleuchtet. Zu den internen Risiken gehören z.B. Konflikte im Team und zu den externen Server Ausfallzeiten beim externen Dienstleister. Hierbei ist zu beachten, dass mit jedem vorangegangenen Projektfortschritt auch Risiken hinzukommen können. Daraufhin werden die Risiken nach ihrer Eintrittswahrscheinlichkeit klassifiziert. Die Einordnung hilft dabei Risiken zu priorisieren. Dabei werden Risiken mit hohen Eintrittswahrscheinlichkeiten genauer betrachtet und in Zukunft im Blick behalten. Zuletzt werden die Risiken nach ihrer Auswirkung eingestuft. Risiken mit hoher Eintrittswahrscheinlichkeit und hohen Auswirkungen können hierbei zum Scheitern des Projektes bzw. zum Nichterfolg führen. Deshalb ist es besonders wichtig diese Art von Risiken nicht nur zu beobachten, sondern kontinuierlich zu messen welche Folgen der Eintritt haben kann.

### 4.2 Grafische Darstellung Risikobewertungsmatrix

Die Risikobewertungsmatrix für das Projekt STH App ist in der folgenden Abbildung (Nr. einfügen) zu sehen.

Für eine App mit diversen online Funktionen ist ein Server Ausfall fatal. Deshalb wurde dieses Risiko als eines der mit am höchsten verbundenen Auswirkungen klassifiziert. Die Eintrittswahrscheinlichkeit bewegt sich hierbei im mittleren Bereich, da nach dem aktuellen Stand der Technik und der Rahmenverträge mit Dienstleistern bzw. Rechenzentren bei einem Ausfall meist auf eine redundante Serverlandschaft ausgewichen werden kann. Ein Datenleck, welches durch einen Angriff auf die Backendsysteme entstehen kann, ist für den Erfolg und gleichzeitig für die Auswirkungen einer Smartphone-App schwerwiegend. Innerhalb der STH App können sensible Daten eingegeben und abgespeichert werden, welche nicht an außenstehende gelangen dürfen. Zudem sind die immensen Kosten

Bild 1: Risikomatrix



Quelle: Quelle Risikomatrix

bezüglich der Vertragsstrafen bei der Nichteinhaltung der DSGVO ein großes finanzielles Risiko. Ein weiteres Risiko für die Applikation sind Urheberrechtsverstöße. Diese können auftreten, indem Nutzer Urheberrechtsgeschützte Inhalte veröffentlichen, welche nicht Ihnen gehören. Eine Begünstigung der Urheberrechtsverstöße könnte auf die Applikation selbst zurückzuführen sein und damit finanziell intensive Vertragsstrafen auslösen. Eine geringe Benutzerakzeptanz kann durch ein vorher schlecht ausgearbeitetes UI/UX Konzept ausgelöst werden. Die Eintrittswahrscheinlichkeit ist hierbei im mittleren Bereich, da es nicht einfach ist den Nutzern ein qualitativ hochwertiges und gut durchdachtes User-Interface zu liefern. Die Auswirkung liegt hierbei ebenso im mittleren Bereich, da das Design schnell angepasst werden kann. Die Budgetüberschreitung ist wie in fast jedem Projekt ein potenzielles Risiko, welches durchaus durch eine schlechte Planung auftreten kann. Sobald die finanziellen Ressourcen ausgeschöpft sind, kann nicht mehr an der Applikation gearbeitet und weiterentwickelt werden. Das könnte unter Umständen zu einem frühzeitigen Scheitern des Projektes führen. In jedem Projektteam ist immer ein gewisses Konfliktpotenzial vorhanden. Dieses Risiko kann jederzeit und vor allem in Hochphasen wie z.B. kurz vor dem Start der Veröffentlichung der Applikation, auftreten. Allerdings ist durch eine gute Projektleitung das Risiko gut einschätzbar und präventiv vermeidbar bzw. zu lösen.

## 5 Implementierung des Chat-Screens

Zu jeder modernen Community Applikation gehört auch die Chatfunktionalität immer dazu. Der Austausch zwischen zwei oder mehreren Menschen ist besonders in der STH App wichtig, um die Kommunikation zwischen Spielern und Managern zu ermöglichen. Die Anforderungen an den Chat-Screen standen schnell fest. Die Funktionalität muss mindestens den gleichen Interaktionsumfang ermöglichen, wie vergleichbare Apps wie z.B. Instagram, Snapchat oder X (ehemalig Twitter).

### 5.1 Anforderungen an den Chat-Screen

Die spezifischen Anforderungen für die Chat-Funktion in der STH App wurden im Projektteam gemeinsam ausgearbeitet. Dabei wurden zunächst vergleichbare Applikationen genauer beleuchtet und die Mindestanforderungen anhand der dort gesichteten Funktionen definiert. Das Chatten in der STH App muss mittels verschlüsselte eins zu eins Verbindungen funktionieren, in der zwei Menschen miteinander in einem Chat-Raum kommunizieren können. Zudem muss es auch die Möglichkeit geben Gruppenchats zu erstellen und mehrere Menschen dazu einzuladen. Auch hierbei müssen die Verbindungen stets verschlüsselt sein. Innerhalb eines Chats muss jeder Nutzer Nachrichten senden und empfangen können. Zu den Nachrichten gehören Textnachrichten, GIFs, Bilder und Videos. Jeder Nutzer muss gesendete und empfangene Nachrichten im Chat sehen können und die Chats müssen mit ihrem kompletten Verlauf beim ein- und ausloggen erhalten und neu geladen werden können.

### 5.2 Erstellung einer Flutter Widget-Seite

Nachdem die Anforderungen definiert wurden, kann nun die Erstellung der Chat-Screen Widget-Seite beginnen. Diese Seiten dienen also Vorbereitung dazu Widgets platzieren und anzeigen zu können. Das Erstellen einer derartigen Seite kann in Flutter durch das Erstellen einer Klasse realisiert werden. Hierbei wird für die Klasse ein kontextbezogener Name eingesetzt und die Art der Widget-Seite definiert werden. Dabei wird zwischen Stateful und Stateless Widgets unterschieden. Bei den Stateful Widget Klassen deren Komponenten sich verändern können. Die Stateless Widgets verändern sich nicht und können sich nicht in der Ansicht aktualisieren. Für den Chat-Screen ist das Stateful Klassenwidget die richtige Wahl, da sich die Anzahl und Inhalte der angezeigten Chats stetig verändern kann.

## 5.3 Integration des Stream-Chat-Flutter Pakets

Die Implementierung einer Chat-Funktion kommt in Social-Media Apps sehr häufig bzw. fast immer vor. Deshalb gibt es hierzu eine große Anzahl an Flutter Paketen, welche sich unter der Seite [pub.dev](https://pub.dev) genauer angeschaut werden können. Ein sehr mächtiges und umfängliches Paket nennt sich hierbei Stream-Chat-Flutter. Dieses Paket bringt eine Anzahl an vorbereiteten Widgets für die Anzeige und Erstellung von Chats mit sich. Darin beinhaltet sind App-Bars, Channel-Screens, Chat-Screens und Chatfunktionen. Die Einbindung dieses Pakets erfordert das Verändern der gesamten Struktur der Anwendung. Die Chat-Funktion stellt einen Hauptbestandteil der Anwendung dar und muss auf den top of Widget-Tree implementiert werden. Das bedeutet, dass das Paket Stream-Chat-Flutter als Erstes in der Anwendung geladen werden muss, um es korrekt auszuführen. Nachdem die Struktur und das Laden der Pakete in der Anwendung angepasst wird, müssen Funktionen geschrieben werden, die das Abrufen von Chats aus dem Stream-Chat-Client ermöglichen. Dabei greift die STH App auf das separate Backend des Paketes zu und authentifiziert den aktuellen Nutzer der Applikation. Nach erfolgreicher Authentifizierung werden die zugehörigen Chatdaten und Channels abgerufen, um den Nutzer alle Chatverläufe korrekt anzeigen zu können.

## 5.4 Benutzerauthentifizierung

Um die Benutzerauthentifizierung durch die Anwendung zu ermöglichen, müssen Zugangsschlüssel bzw. Tokens bereitstehen. Diese Tokens werden durch den Aufruf durch Firebase generiert und der jeweilige Nutzer der Anwendung registriert. Nach erfolgreicher Übermittlung der Tokens, kann die STH App nun sicherstellen, dass es sich um den angemeldeten Nutzer handelt und alle Chatverläufe inklusive persönlicher Daten laden.

## 5.5 Implementierung der Chat-Widgets

Nach der erfolgreichen Benutzerauthentifizierung folgt der letzte Schritt. Die verschiedenen Chats des Nutzers und die Chatverläufe können mittels der Chat-Widgets aus dem Paket Stream-Chat-Flutter angezeigt werden. Das Paket beinhaltet einen vollständigen Channel-Screens. Das ist das Widget, welches zuerst beim Einstieg in die Chat-Funktionen angezeigt wird. Analog zu anderen bekannten Applikation wird hierbei eine Übersicht über alle aktiven Chats des Nutzers angezeigt. Daraufhin folgt der Channel-Screens. Dieser stellt den einzelnen Chat, auf dem vom Channel-Screens aus geklickt wird, dar. Im Chat-Screen ist der gesamte Chatverlauf aus diesem Chat des Nutzers zu sehen. Zudem kann hier durch das Textfeld am unteren Ende der Seite neue Nachrichten durch den Nutzer eingegeben werden und an den Chatpartner versendet werden.

Die Implementierung des Channel-Screens erfolgt als Erstes. Hierfür wird eine eigene Klasse erstellt und das Chat-Screen Widget aus dem Paket wird auf dieser Seite platziert. Nun muss noch das richtige Routing definiert werden. Das stellt sicher, dass die verschiedenen Chats geladen werden können. Die Liste der verfügbaren Chats wird geladen und am oberen Ende der Seite wird die App-Bar mit dem Namen Chats angezeigt.

Nach der Integration des Channel-Screen-Widgets kann nun das Chat-Screen-Widget folgen. Für das Chat-Screen-Widget wird ebenso eine neue Klasse und Seite erstellt, worauf das Widget platziert werden kann. Nach der Platzierung wird durch das korrekte Routing sichergestellt, dass der richtige Chat vom angemeldeten User geöffnet wird. Durch die nahtlose Integration des Paketes sind alle Funktionen automatisch verfügbar. Das Versenden und empfangen von Nachrichten ist bereits möglich und das Reagieren auf Nachrichten ebenso. Zu normalen Textnachrichten können auch Bilder, GIFs und Emojis versendet werden. Die Grundfunktionalitäten einer ausgereiften Social-Media App stehen nun mit der Chat-Funktion.

Nun ist der letzte Schritt die Anwendung auf die korrekte Wegweisung bzw. Routing anzulernen. Dabei wird in der `main.dart`, welches die Hauptdatei darstellt und dabei auch die Strukturen für die gesamte Anwendung festlegt, eingearbeitet. Die neuen Screens müssen den gesamten Routing Optionen der Anwendung hinzugefügt werden. Die dabei verwendeten Klassen werden dabei referenziert. Innerhalb der Datei `CustomAppBar` wird nun für den Chat-Button das korrekte Routing hinterlegt. Dabei wird auch festgelegt in welchen Bildschirmen der Chat-Button sichtbar ist und in welchen nicht. Innerhalb des Chat-Screens verschwindet der Chat-Button, da der Anwender sich bereits in der Chat-Funktionalität aufhält. In allen anderen Ansichten ist der Button sichtbar und das Routing erfolgt beim auslösen. Nun ist der Channel-Screen und der Chat-Screen in der Anwendung vollständig mit korrektem Routing implementiert.