



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

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

STH-App *Sport Talent Hub App*

IDE *Integrierte Entwicklungsumgebung*

iOS *Internetwork Operating System*

LaTeX *Lamport TeX*

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.

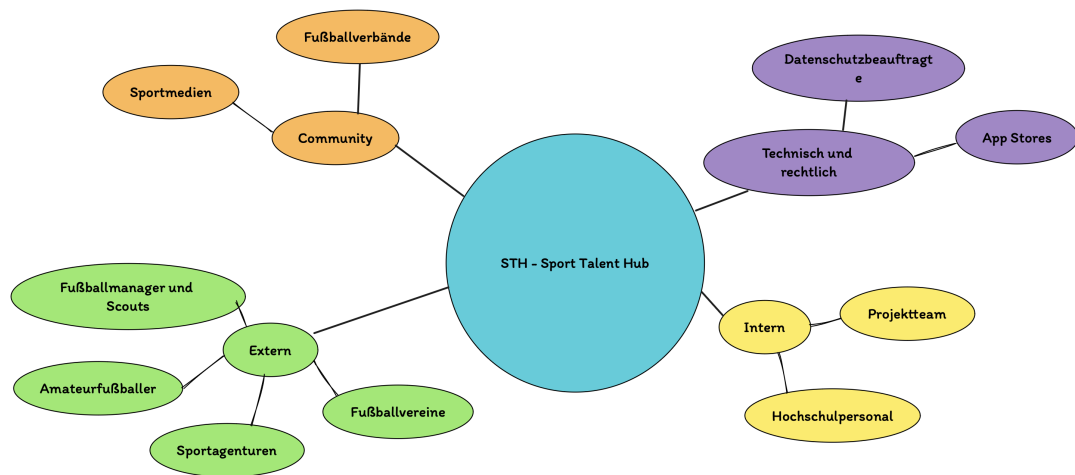
3.1 Vorgehen Stakeholderanalyse

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

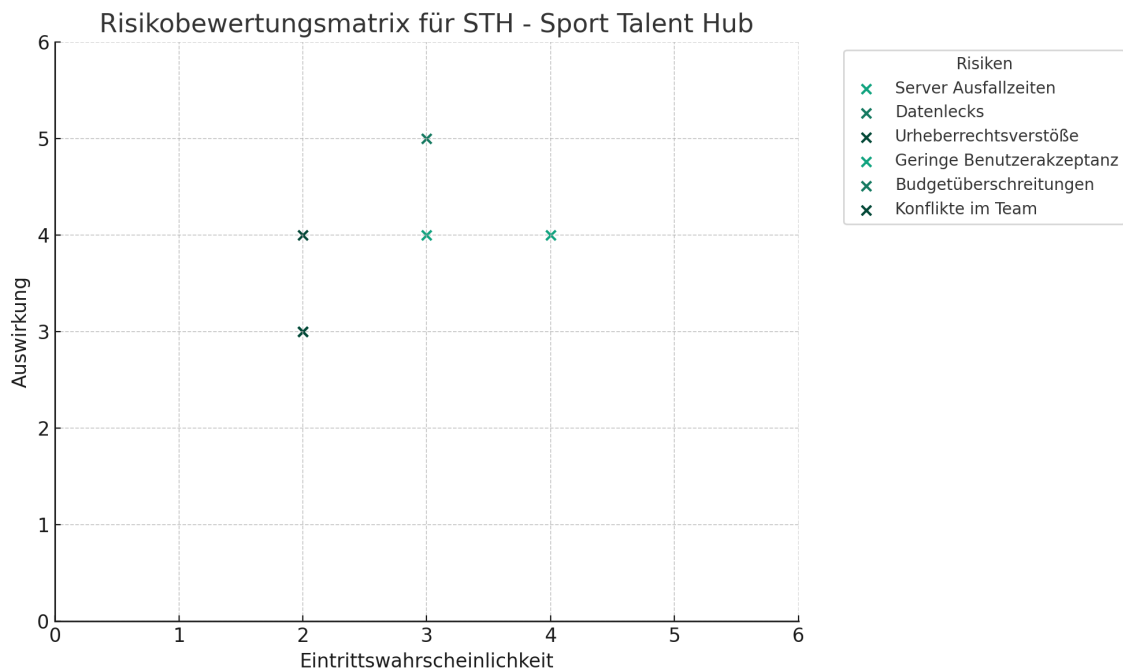
Um eine Risikobewertungsmatrix erstellen und veranschaulichen zu können benötigt es einige Schritte. Zunächst müssen die Risiken erkannt werden, die im Projekt auftreten können. Dabei werden sowohl interne als auch externe Faktoren beleuchtet. Zu den internen Risiken gehören z.B. Konflikte im Team und 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 Produktanforderungen und technische Architektur

In diesem Kapitel werfen wir einen umfassenden Blick auf die Produktanforderungen und die technische Architektur der STH-App. Die STH-App bietet weit mehr als nur das Teilen von Inhalten. Sie fungiert als Social-Media-Plattform, auf der Sportmanager und Sportler miteinander kommunizieren und Fähigkeiten sowie personenbezogene Profildaten teilen können. Daher ist es von entscheidender Bedeutung, dass die Produktanforderungen einzelner Funktionen und Features separat zur Implementierung bereitgestellt werden, um den unterschiedlichen Anforderungen und Erwartungen beider Gruppen gerecht zu werden. Insbesondere mit der großen Anzahl an Funktionen ermöglicht die STH-App den Nutzern, ihre sportliche Karriere zu teilen und sich dabei mit anderen Benutzern vernetzen können. Der Home-Screen der STH-App bietet einen umfassenden Überblick über die sportlichen Aktivitäten und Leistungen durch Posts der jeweiligen Benutzer. Auf diese Weise können Manager beispielsweise die Profile von Athleten verfolgen. Gleichzeitig ermöglicht er den Sportlern, ihre Fähigkeiten zu veröffentlichen und sich mit anderen Athleten und Sportmanagern anderer Sportvereine zu vernetzen. Der Chat-Screen hingegen bietet für Sportmanager und Athleten einen zentralen Bereich für den direkten Austausch von Textnachrichten über Chatkanäle. Dies ermöglicht eine reibungslose Kommunikation und Zusammenarbeit, indem wichtige Informationen schnell und effizient ausgetauscht werden können, um die gemeinsamen Vereinbarungen und Ziele zu erreichen. Auf dem Search-Screen können Benutzer nach spezifischen personenbezogenen Daten oder Hashtags suchen. So wird beispielsweise das Auffinden neuer Benutzer oder das Suchen nach individuellen Informationen erleichtert, die den eigenen Interessen entsprechen. Beim Profile-Screen können Benutzer die digitale Visitenkarte eines jeden Sportlers oder Managers einsehen. Diese enthält personenbezogene Daten sowie eine Beschreibung der Person, Bilder, Videos und Qualifikationen und ermöglicht es den Benutzern, sich ein umfassendes Bild von anderen Nutzern zu machen und potenzielle Verbindungen zu knüpfen oder Zusammenarbeit zu erleichtern. Die Leistung der STH-App spielt eine ebenso wichtige Rolle, um sicherzustellen, dass die Benutzer ein reibungsloses Erlebnis genießen können. Dabei erfordert es eine effiziente Programmierung und Optimierung sowohl auf dem Frontend als auch auf dem Backend der App. Der Loading-Screen unterstützt dabei, sämtliche Initialisierungsfeatures der STH-App aufzurufen und damit alle verbundenen Services zu starten und zu initialisieren. Neben den Produktanforderungen ist auch die technische Architektur der App von entscheidender Bedeutung. Hierbei spielt die technische Architektur einer App eine entscheidende Rolle, da sie die Grundlage für Effizienz und Skalierbarkeit bildet. Ein sorgfältig ausgewähltes Set von Technologien und Plattformen ist daher unerlässlich, um sicherzustellen, dass die Anwendung nicht nur reibungslos funktioniert, sondern auch die sich ständig wandelnden Anforderungen der Benutzer erfüllen kann. Als Entwicklungsplattform dient Visual

Studio Code, eine leistungsstarke IDE, die speziell für die Entwicklung von Flutter-Apps optimiert ist. Visual Studio Code bietet eine Fülle von Funktionen und Erweiterungen, die den Entwicklungsprozess beschleunigen und vereinfachen. Die klare Definition der Produktanforderungen sowie die Festlegung der Arbeitspakete über GitHub-Commits, in denen die Aufgabenstellung detailliert beschrieben wird, bilden zusammen mit der soliden technischen Architektur das Fundament für eine effektive Zusammenarbeit in der Entwicklung und Dokumentation des Projekts. Flutter selbst fungiert als Cross-Platform-Framework für die Entwicklung des Frontends als auch der Schnittstelle zum Backend der STH-App. Die Verwendung von Flutter bietet eine Reihe von Vorteilen, darunter eine hohe Leistung, schnelle Entwicklung und einfache Wartung. Für das Backend der STH-App wird Firebase genutzt, eine Plattform von Google, die eine breite Palette von Diensten für die Entwicklung von Web- und Mobile-Apps bietet. Mit Funktionen wie Echtzeitdatenbanken, Authentifizierungsdiensten und Cloud-Speichern bietet Firebase eine robuste Lösung für die Anforderungen der STH-App.

6 Teilprojekte und Arbeitspakete

In diesem Kapitel liegt der Fokus auf den Arbeitspaketen, die für die Entwicklung der STH-App von entscheidender Bedeutung sind. Es wird zunächst erläutert, welche Aufgabenstellung jedes Arbeitspaket initialisiert, wobei sowohl die Verantwortlichen des Arbeitspakets als auch die damit verbundenen Ressourcen betrachtet werden.

Definition der Arbeitspakete:

Arbeitspaket 1 vom 08.03.2024 bis 11.03.2024

- Bezeichnung: Initial Setup Flutter
- Verantwortlichen: Das Team
- Ressourcen: VS Code, Flutter SDK, Smartphone Emulatoren
- Aufgabenstellung: Einrichtung der Entwicklungsumgebung für Flutter

Arbeitspaket 2 vom 08.03.2024 bis 11.03.2024

- Bezeichnung: Initial Setup LaTeX
- Verantwortlichen: Das Team
- Ressourcen: VS Code inklusive LaTeX-Plugins, Container-Plattform (Docker)
- Aufgabenstellung: Einrichtung der Entwicklungsumgebung für LaTeX

Arbeitspaket 3 vom 08.03.2024 bis 11.03.2024

- Bezeichnung: Initial Setup GitHub
- Verantwortlichen: Das Team
- Ressourcen: Git-Client, GitHub-Konto
- Aufgabenstellung: Erstellung des Repositories auf GitHub

Arbeitspaket 4 vom 13.03.2024 bis 14.03.2024

- Bezeichnung: Project Roadmap
- Verantwortlichen: Das Team
- Ressourcen: Brainstorming
- Aufgabenstellung: Definition der App Funktionalitäten

Arbeitspaket 5 vom 14.03.2024 bis 15.03.2024

- Bezeichnung: Project Structure
- Verantwortlichen: Herr Jonas Waigel
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Erstellung der ersten Projektstruktur in Visual Studio Code/Flutter

Arbeitspaket 6 vom 16.03.2024 bis 23.03.2024

- Bezeichnung: Home Screen
- Verantwortlichen: Herr Jonas Waigel
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung des Home-Screens inklusive Menüleiste am unteren Bildschirmrand

Arbeitspaket 7 vom 18.03.2024 bis 01.04.2024

- Bezeichnung: Chat Screen
- Verantwortlichen: Herr Hasan Deveci
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung des Chat-Screens

Arbeitspaket 8 vom 21.03.2024 bis 04.04.2024

- Bezeichnung: Account Profile
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung des Account-Profile-Screens inklusive Avatar und personenbezogene Daten

Arbeitspaket 9 vom 23.03.2024 bis 06.04.2024

- Bezeichnung: Search Screen
- Verantwortlichen: Herr Vatsegkan Zournatsidis
- Ressourcen: ChatGPT 4.0
- Aufgabenstellung: Implementierung des Search-Screens

Arbeitspaket 10 vom 23.03.2024 bis 31.03.2024

- Bezeichnung: Global Menu Bar
- Verantwortlichen: Herr Jonas Waigel
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung einer globalen Menüleiste am unteren Bildschirmrand für alle Pages

Arbeitspaket 11 vom 24.03.2024 bis 07.04.2024

- Bezeichnung: Initial Setup Backend
- Verantwortlichen: Herr Jonas Waigel
- Ressourcen: VS Code, Flutter Framework, Backend-Plattform (Firebase)
- Aufgabenstellung: Backend-Initialisierung (Firebase)

Arbeitspaket 12 vom 24.03.2024 bis 31.03.2024

- Bezeichnung: Profile Edit Button
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung eines Bearbeitungsbuttons für den Account-Profile-Screen

Arbeitspaket 13 vom 24.03.2024 bis 31.03.2024

- Bezeichnung: Logo Design
- Verantwortlichen: Herr Vatsegkan Zournatsidis
- Ressourcen: ChatGPT 4.0
- Aufgabenstellung: Gestaltung eines Logos für die STH-APP

Arbeitspaket 14 vom 25.03.2024 bis 04.04.2024

- Bezeichnung: Loading Screen
- Verantwortlichen: Herr Vatsegkan Zournatsidis
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung einer Ladebildschirm-Funktion beim Start der STH-App

Arbeitspaket 15 vom 27.03.2024 bis 06.04.2024

- Bezeichnung: GetStream Chat
- Verantwortlichen: Herr Hasan Deveci
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung von GetStream-Chat

Arbeitspaket 16 vom 27.03.2024 bis 28.03.2024

- Bezeichnung: Chat App Bar
- Verantwortlichen: Herr Jonas Waigel
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Anpassung der App-Leiste auf dem Chat-Bildschirm

Arbeitspaket 17 vom 29.03.2024 bis 04.04.2024

- Bezeichnung: Logo Loading Screen
- Verantwortlichen: Herr Vatsegkan Zournatsidis
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung eines Ladebildschirms mit einem Logo

Arbeitspaket 18 vom 31.03.2024 bis 07.04.2024

- Bezeichnung: Backend Config
- Verantwortlichen: Herr Jonas Waigel
- Ressourcen: VS Code, Flutter Framework, Backend-Plattform (Firebase)
- Aufgabenstellung: Implementierung der Backend-Konfiguration (Firebase) und deren Methoden

Arbeitspaket 19 vom 01.04.2024 bis 07.04.2024

- Bezeichnung: Custom Page Routing
- Verantwortlichen: Herr Jonas Waigel
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung einer benutzerdefinierten Seitenroute beim Wechsel von den Pages

Arbeitspaket 20 vom 02.04.2024 bis 08.04.2024

- Bezeichnung: Chat UI Adjustments
- Verantwortlichen: Herr Hasan Deveci
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Anpassung des Chat-Screens und der benutzerdefinierte App-Leiste

Arbeitspaket 21 vom 02.04.2024 bis 08.04.2024

- Bezeichnung: Channel Adjustment
- Verantwortlichen: Herr Hasan Deveci
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Anpassung des Channels im Chat-Screen

Arbeitspaket 22 vom 02.04.2024 bis 03.04.2024

- Bezeichnung: Latex Docs Update
- Verantwortlichen: Herr Jonas Waigel
- Ressourcen: VS Code
- Aufgabenstellung: Aufgabenstellung: Dokumentationsstruktur in LaTeX aktualisieren

Arbeitspaket 23 vom 03.04.2024 bis 10.04.2024

- Bezeichnung: Local Storage Setup
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung eines lokalen Speichers für den Account-Profile-Screen

Arbeitspaket 24 vom 03.04.2024 bis 10.04.2024

- Bezeichnung: Save/Load Functions
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung einer Funktion zum Speichern/Laden für den Account-Profile-Screen

Arbeitspaket 25 vom 03.04.2024 bis 10.04.2024

- Bezeichnung: RegEx Guidelines
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung von RegEx-Richtlinien in Bezug auf personenbezogene Daten

Arbeitspaket 26 vom 03.04.2024 bis 15.04.2024

- Bezeichnung: Hovering Function
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung einer Hovering-Funktion für personenbezogene Daten

Arbeitspaket 27 vom 03.04.2024 bis 15.04.2024

- Bezeichnung: Profile UI
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung des Profile-Screens mit Avatar, Bilder und Mediathek

Arbeitspaket 28 vom 10.04.2024 bis 18.04.2024

- Bezeichnung: Media Upload
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung von Funktionen zum Auswählen und Hochladen von Bildern und Videos

Arbeitspaket 29 vom 10.04.2024 bis 18.04.2024

- Bezeichnung: Profile Local Storage
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung eines lokalen Speichers für den Profile-Screen

Arbeitspaket 30 vom 10.04.2024 bis 18.04.2024

- Bezeichnung: Profile Save/Load
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung einer Funktion zum Speichern/ Laden für den Profile-Screen

Arbeitspaket 31 vom 10.04.2024 bis 11.04.2024

- Bezeichnung: App Bar Navigation
- Verantwortlichen: Herr Jonas Waigel
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Anpassung der App-Leiste für die Navigation und das Verhalten beim Seitenwechsel

Arbeitspaket 32 vom 13.04.2024 bis 19.04.2024

- Bezeichnung: Shared Preferences
- Verantwortlichen: Herr Jonas Waigel
- Ressourcen: VS Code, Flutter Framework, Backend-Plattform (Firebase)
- Aufgabenstellung: Verwendung von Shared Preferences zusammen mit Firebase-Testdaten

Arbeitspaket 33 vom 14.04.2024 bis 15.04.2024

- Bezeichnung: Page Linking
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Verlinkung der Pages von Profile-Screen und Account-Profile-Screen

Arbeitspaket 34 vom 16.04.2024 bis 17.04.2024

- Bezeichnung: Loading Screen Update
- Verantwortlichen: Herr Vatsegkan Zournatsidis
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Aktualisierung des Loadingscreens

Arbeitspaket 35 vom 16.04.2024 bis 25.04.2024

- Bezeichnung: Start Page Logo
- Verantwortlichen: Herr Vatsegkan Zournatsidis
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Hinzufügen vom Logo auf die Startseite

Arbeitspaket 36 vom 21.04.2024 bis 28.04.2024

- Bezeichnung: Media Button
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung eines Buttons für das Hochladen von Bilder und Videos

Arbeitspaket 37 vom 21.04.2024 bis 01.05.2024

- Bezeichnung: Avatar Transmission
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code, Flutter Framework, Backend-Plattform (Firebase)
- Aufgabenstellung: Implementierung von Funktionen zur Übertragung vom Avatar zum Backend

Arbeitspaket 38 vom 25.04.2024 bis 05.05.2024

- Bezeichnung: Media Transmission
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code, Flutter Framework, Backend-Plattform (Firebase)
- Aufgabenstellung: Implementierung von Funktionen zur Übertragung von Bilder und Videos an das Backend

Arbeitspaket 39 vom 27.04.2024 bis 28.04.2024

- Bezeichnung: Page Navigation
- Verantwortlichen: Herr Jonas Waigel
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Anpassung der Seiten-Navigation und Entfernung von Animationen beim Wechsel

Arbeitspaket 40 vom 28.04.2024 bis 29.04.2024

- Bezeichnung: Latex Docs Update
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code
- Aufgabenstellung: Dokumentationsstruktur in Latex aktualisiert

Arbeitspaket 41 vom 29.04.2024 bis 05.04.2024

- Bezeichnung: Chat Routing
- Verantwortlichen: Herr Hasan Deveci
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung der Seitenroute vom Chat zum Channelscreen

Arbeitspaket 42 vom 29.04.2024 bis 05.04.2024

- Bezeichnung: Profile Navigation
- Verantwortlichen: Herr Hasan Deveci
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung der Navigationsroute vom Chat-Screen zum Profile-Screen

Arbeitspaket 43 vom 30.04.2024 bis 10.05.2024

- Bezeichnung: Stream Headers
- Verantwortlichen: Herr Hasan Deveci
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung des Stream-Channel-Headers

Arbeitspaket 44 vom 30.04.2024 bis 10.05.2024

- Bezeichnung: Username Update
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung der Aktualisierung des Benutzernamens auf dem Profile-Screen

Arbeitspaket 45 vom 30.04.2024 bis 07.05.2024

- Bezeichnung: Action Button
- Verantwortlichen: Herr Hasan Deveci
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung einer Aktionsschaltfläche mit Benutzerintegration

Arbeitspaket 46 vom 30.04.2024 bis 07.05.2024

- Bezeichnung: Test User Addition
- Verantwortlichen: Herr Hasan Deveci
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung neuer Chat-User als Testdaten

Arbeitspaket 47 vom 05.05.2024 bis 15.05.2024

- Bezeichnung: Hashtag Implementation
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung von Hashtags im Profile-Screen

Arbeitspaket 48 vom 06.05.2024 bis 15.05.2024

- Bezeichnung: Hashtag Storage
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung eines lokalen Speichers für die Hashtags

Arbeitspaket 49 vom 06.05.2024 bis 15.05.2024

- Bezeichnung: Hashtag Transmission
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Implementierung von Funktionen zur Übertragung von Hashtags an das Backend

Arbeitspaket 50 vom 07.05.2024 bis 20.05.2024

- Bezeichnung: Major Bug Fixes
- Verantwortlichen: Herr Fitim Makolli
- Ressourcen: VS Code, Flutter Framework
- Aufgabenstellung: Fehlerbehebungen für den Profile-Screen und Account-Profile-Screen

7 Meilensteinplanung

Im Rahmen dieses Projekts war es erforderlich, eine detaillierte Planung und Organisation der Arbeitspakete vorzunehmen. Um dies zu erreichen, haben wir die Tabelle mit den Arbeitspaketen analysiert und ein Gantt-Diagramm erstellt. Das Gantt-Diagramm bietet eine visuelle Darstellung der zeitlichen Abfolge und Dauer der verschiedenen Aufgaben, was die Projektplanung und -kontrolle erleichtert.

7.1 Vorgehensweise zur Erstellung des Gantt-Diagramms

Zunächst haben wir die Daten aus der Tabelle extrahiert und aufbereitet. Die Tabelle enthielt drei wesentliche Spalten: Arbeitspaket, Beschreibung und Dauer. Diese Daten wurden in ein geeignetes Format gebracht, um sie in das Gantt-Diagramm-Tool einzugeben.

Zur Erstellung des Gantt-Diagramms wurde die Website onlinegantt.com genutzt. Diese Plattform bietet eine benutzerfreundliche Oberfläche zur Eingabe und Visualisierung von Projektplänen.

Für jedes Arbeitspaket wurde das Startdatum entsprechend der Tabelle eingegeben. Die Beschreibung der Aufgaben wurde als Titel für die jeweiligen Arbeitspakete verwendet. Die Dauer der Aufgaben wurde in Tagen oder Wochen entsprechend der Tabelle angegeben.

Nachdem alle Daten eingegeben wurden, generierte die Plattform das Gantt-Diagramm. Dieses Diagramm ermöglicht es, die zeitliche Abfolge der Aufgaben zu visualisieren und Überschneidungen oder Abhängigkeiten zwischen den Aufgaben zu erkennen.

7.2 Analyse der Zeitplanung

7.2.1 Gesamtübersicht

Die Erstellung des Gantt-Diagramms zeigte eine klare Struktur des Projekts, beginnend am 08. März 2024 und endend am 30. April 2024. Die Aufgaben sind logisch und sequenziell angeordnet, was die Nachverfolgung und Kontrolle erleichtert.

7.2.2 Dauer und Überlappungen

Einige Aufgaben, wie die Einrichtung des Repositories auf GitHub oder die Erstellung der Projektstruktur, hatten eine kurze Dauer von nur einem Tag. Andere Aufgaben, wie

die Implementierung des Chat-Screens oder die Anpassung der Seiten-Navigation, dauerten mehrere Wochen. Diese längeren Aufgaben wurden parallel zu kürzeren Aufgaben geplant, um die Gesamtzeit des Projekts effizient zu nutzen.

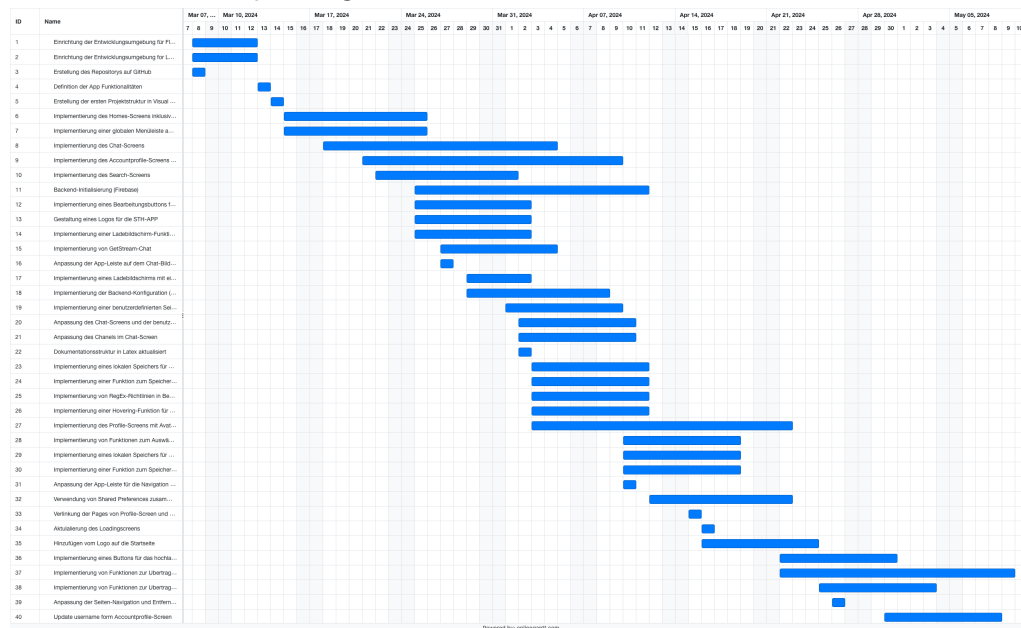
7.2.3 Wichtige Meilensteine

Einige wesentliche Meilensteine des Projekts sind: Die ersten beiden Wochen des Projekts waren der Einrichtung der Entwicklungsumgebungen und der Definition der App-Funktionalitäten gewidmet. Der Großteil des Projekts, von Mitte März bis Ende April, bestand aus der Implementierung verschiedener Funktionen und der Backend-Integration. Die letzten Tage des Projekts wurden für Feinanpassungen und Updates, wie die Aktualisierung des Loadingscreens und das Update des Usernames auf dem Accountprofile-Screen, verwendet.

7.2.4 Risikobewertung

Durch das Gantt-Diagramm konnten potenzielle Engpässe und Risiken identifiziert werden. Aufgaben, die die kritische Pfade darstellen, wurden besonders beachtet, um Verzögerungen zu vermeiden.

Bild 1: Meilensteinplanung



Quelle: <https://www.onlinegantt.com>

8 Ladebildschirm

Der Ladebildschirm ist ein essenzieller Bestandteil der Benutzererfahrung unserer STH-App (SportTalentHub). Er erscheint, bevor der Nutzer die Hauptfunktionen der App nutzen kann, und dient als Zwischenbildschirm, um dem Nutzer zu signalisieren, dass die App lädt.

8.1 Ziele des Ladebildschirms

Der Ladebildschirm hat mehrere wichtige Funktionen. Er zeigt dem Nutzer visuelles Feedback, dass die App aktiv ist und lädt, wodurch eine bessere Benutzererfahrung gewährleistet wird. Der Ladebildschirm wurde so gestaltet, dass er genau vier Sekunden lang angezeigt wird. Dies gibt der App ausreichend Zeit, um die notwendigen Daten und Ressourcen im Hintergrund zu laden. Es ist entscheidend, dass nach Ablauf der vier Sekunden die Benutzer zur Startseite (Homepage) der App navigiert werden und nicht zu anderen Seiten wie der Profil- oder Chatseite.

8.2 Gestaltung des Ladebildschirms

Für die Gestaltung des Ladebildschirms waren mehrere Schritte notwendig. Das Logo wurde mit Canva erstellt und musste den Charakter und die Zielgruppe der App widerspiegeln. Es wurde darauf geachtet, dass das Design für eine SportTalentHub-App geeignet ist. Die Inspiration für das Logo wurde aus verschiedenen Quellen, wie der NFL, gezogen. Die Farben und das Design sollten sportlich und ansprechend sein. Nach der Erstellung wurde das Logo transparent gemacht, um es optimal in den Ladebildschirm integrieren zu können.

Bevor das Logo endgültig in die App integriert wurde, wurde es im Rahmen eines wöchentlichen Meetings präsentiert. Das positive Feedback der Teammitglieder bestätigte die Eignung des Logos, sodass es anschließend in den Ladebildschirm eingefügt wurde.

Das Logo wurde im Ladebildschirm implementiert und ein animiertes Symbol hinzugefügt, das die Ladebewegung anzeigt. Es wurde darauf geachtet, dass nach dem Ablauf der vier Sekunden der Nutzer zur Startseite navigiert wird. Dies wurde durch entsprechende Programmierung in Flutter sichergestellt.

8.3 Änderungen am App-Logo

Neben der Erstellung und Implementierung des Logos für den Ladebildschirm war es auch notwendig, das App-Logo selbst zu ändern. Das Ändern des App-Logos war technisch

weniger anspruchsvoll. Es erforderte lediglich, das neue Logo in den entsprechenden Bereich des App-Projekts einzufügen.

Wir entschieden uns außerdem, das Logo auch auf der Startseite der App anzuzeigen. Hierzu wurde ein Code in die Startseite geschrieben, der diese Funktion realisiert.

9 Suchbildschirm

Der Suchbildschirm ist eine zentrale Funktion unserer STH-App (SportTalentHub). Er ermöglicht es den Nutzern, sowohl Sportlern als auch Sportmanagern, gezielt nach bestimmten Personen anhand von Hashtags oder Namen zu suchen.

9.1 Ziele des Suchbildschirms

Das Hauptziel des Suchbildschirms ist es, den Nutzern eine effiziente Möglichkeit zu bieten, Spieler oder Manager zu finden. Dies kann durch die Eingabe von Hashtags oder Namen erfolgen. Die Suchfunktion ist über das Suchsymbol auf dem Startbildschirm zugänglich.

9.2 Integration mit Firebase

Die Suchfunktion ist eng mit Firebase verbunden. Firebase spielt eine entscheidende Rolle, da dort die Profildaten der Nutzer gespeichert sind. Dadurch können Nutzer problemlos nach anderen Nutzern suchen. Wenn ein Suchbegriff, sei es ein Hashtag oder ein Name, eingegeben wird, erscheinen die gesuchten Personen auf dem Suchbildschirm. Bei Auswahl einer Person wird der Nutzer zu deren Profil weitergeleitet.

9.3 Wichtigkeit der Suchfunktion

Die Suchfunktion ist für unsere App von großer Bedeutung. Sie ermöglicht es Sportmanagern, gezielt nach bestimmten Merkmalen oder Fähigkeiten zu suchen und schnell und effektiv Ergebnisse zu erhalten. Dies verbessert die Benutzererfahrung und erhöht die Effizienz der App.

9.4 Zukünftige Erweiterungen

Zukünftige Erweiterungen der Suchfunktion könnten die Speicherung früher gesuchter Personen im Suchbereich umfassen. Außerdem wäre es möglich, die Suche zu erweitern, sodass nicht nur nach Hashtags und Namen gesucht werden kann, sondern auch nach weiteren Informationen oder Merkmalen.