**Hochschule Esslingen**  
Sommersemester 2023



**Projekt Sport Challenge**  
Softwareprojekt für doubleSlash

Projektteam:   
Jason Patrick Duffy (jaduit00@hs-esslingen.de)  
Robin Hackh (rohait02@hs-esslingen.de)  
Tom Nguyen Dinh (tongit00@hs-esslingen.de)  
Mason Schönherr (mascit43@hs-esslingen.de)

Betreuer: Herr Prof. Dr. Nitzsche  
Kunden: doubleSlash Net-Business GmbH

Stand: 12. April 2023

Inhaltsverzeichnis

[1 Einführung 3](#_Toc132226817)

[1.1 Projektbeschreibung 3](#_Toc132226818)

[1.2 Anforderungen 3](#_Toc132226819)

[1.2.1 Beschränkungen 4](#_Toc132226820)

[1.3 Qualitätsziele 4](#_Toc132226821)

[1.4 Risiken 4](#_Toc132226822)

[1.5 Zielgruppe 5](#_Toc132226823)

[1.6 Personas 5](#_Toc132226824)

[2 Projektmanagement 7](#_Toc132226825)

[2.1 Agiles Projektmanagement 7](#_Toc132226826)

[2.1.1 Terminplan 7](#_Toc132226827)

[2.1.2 Rollenverteilung 7](#_Toc132226828)

[2.1.3 Definition of Ready 8](#_Toc132226829)

[2.1.4 Definition of Done 8](#_Toc132226830)

[2.2 Projektverwaltung 8](#_Toc132226831)

[2.2.1 User Story Regeln 8](#_Toc132226832)

[2.3 Versionsverwaltung 9](#_Toc132226833)

[2.3.1 Commit Regeln 9](#_Toc132226834)

[2.3.2 Branch Regeln 10](#_Toc132226835)

[2.4 Teamkommunikation 11](#_Toc132226836)

[2.5 Implementierung 11](#_Toc132226837)

[2.5.1 Testing Policy 12](#_Toc132226838)

[3 Funktionsumfang 13](#_Toc132226839)

[3.1 User Stories 13](#_Toc132226840)

[4 UI-Entwürfe 21](#_Toc132226841)

[5 Architektur 22](#_Toc132226842)

[5.1 Lizenzen 23](#_Toc132226843)

[5.1.1 Projekt 23](#_Toc132226844)

[5.1.2 React 23](#_Toc132226845)

[5.1.3 Java Spring Boot 23](#_Toc132226846)

[5.1.4 Keycloak 23](#_Toc132226847)

[5.1.5 PostgreSQL 23](#_Toc132226848)

[5.2 Diagramme 23](#_Toc132226849)

[5.2.1 ER-Diagramm 23](#_Toc132226850)

[5.2.2 Klassendiagramm 24](#_Toc132226851)

[5.2.3 Ablaufdiagramm 24](#_Toc132226852)

[6 Aufwandschätzung 26](#_Toc132226853)

[Quellenverzeichnis 27](#_Toc132226854)

[Abbildungsverzeichnis 27](#_Toc132226855)

# 1 Einführung

Das Sport Challenge Projekt wird im Auftrag der Firma doubleSlash Net-Business GmbH durchgeführt. Dieses Dokument dokumentiert die Umsetzung des Projekts und die Regeln, die wir für dieses Projekt festgelegt haben.

## 1.1 Projektbeschreibung

Bei der doubleSlash Rad- und Laufaktion können Benutzer zusammen mit einem Team an befristeten Challenges teilnehmen, bei denen ihnen und ihrem Team ihre zurückgelegte Strecke in Punkten gutgeschrieben wird. Je nach Sportart (Laufen, Fahrradfahren, etc.) wird zur Strecke eine unterschiedliche Gewichtung als Faktor verwendet.

Am Ende der Challenge erhält das Gewinnerteam eine Belohnung, wie ein kostenloses Abendessen, und der gesammelte Kilometerstand wird mit einem Faktor zu einer wohltätigen Spende umgerechnet.

Die Dateneingabe erfolgt auf Vertrauensbasis und benötigt keine weitere Verifikation.

Folgende Features sind dabei von doubleSlash gewünscht:

* Verwalten von Sportarten mit unterschiedlicher Gewichtung
* Verwalten von Teilnehmern und Teams
* Verwalten von Bonusaktionen (z.B. Tage, an denen andere Gewichtungen gelten)
* Hinzufügen von Daten
* Benachrichtigungen bei Updates und Bonusaktionen
* Übersicht aller Teilnehmer und Teams und deren Platzierung
* Auswertung aller Eingaben und deren Ergebnissen
* Authentifizieren der Teilnehmer über SAML oder Windows AD

Folgende Features sind „nice to have”:

* API-Endpunkte zum Anbinden eines Datenimporters
* Einbindung der Strava-API
* Benutzergruppen mit unterschiedlichen Rechten

Ziel ist es, eine neue Full-Stack-Anwendung zum Koordinieren der firmeninternen Rad- und Laufaktionen umzusetzen und die zuvor verwendete Excel-Tabelle abzulösen.

## 1.2 Anforderungen

In diesem Abschnitt werden die von doubleSlash definierten funktionalen und nicht-funktionalen Anforderungen definiert und erläutert.

|  |  |  |
| --- | --- | --- |
| **ID** | **Anforderung** | **Erläuterung** |
| F1 | Im Browser aufrufbar | Die Anwendung soll im Web-Browser aufgerufen und vollständig verwaltet werden können. |
| F1.1 | Responsive | Da die Anwendung im Web-Browser angezeigt werden soll, muss sie sich responsiv dem Browserfenster anpassen und auch auf Mobiltelefonen problemlos funktionieren. |
| F2 | Erweiterbarkeit | Die Anwendung soll zukünftig erweiterbar, beispielsweise durch das Hinzufügen neuer Sportarten oder Anbinden einer Drittanbieter-API, sein. |
| F3 | Containerized | Die gesamte Anwendung soll in einem Docker-Container laufen. |
| NF1 | Agile | Das Projekt soll mithilfe agiler Methoden durchgeführt werden. |

### 1.2.1 Beschränkungen

Aus den definierten Anforderungen ergeben sich folgende Beschränkungen.

Die Anwendung sollte…

* …Plattformunabhängig über den Webbrowser aufgerufen werden können.
* …in einem Docker-Container laufen.
* …eine Lizenz verwenden, die dem Kunden das Verändern und Einsetzen ermöglicht.

## 1.3 Qualitätsziele

In diesem Abschnitt werden die Qualitätsziele nach ISO 25010[[1]](#footnote-2) beschreiben und einige dazugehörige Szenarien definiert.

|  |  |  |  |
| --- | --- | --- | --- |
| **Kategorie** | **Qualität** | **Beschreibung** | **Szenario** |
| Compatibility | Interoperability | Wenn möglich, sollte eine Schnittstelle zum Import von Daten bereitgestellt werden. | C1 |
| Usability | Learnability | Das System muss intuitiv sein und keine vorherige Einführung benötigen. | U1 |
| Usability | Operability | Die Dateneingabe sollte nicht zu viel Zeit in Anspruch nehmen. |  |
| Security | Integrity | Der Zugriff auf die Anwendung soll auf die Mitarbeiter von doubleSlash begrenzt sein. |  |
| Maintainability | Modularity | Das System muss einfach erweiterbar sein. | M1 |
| Portability | Installability | Die gesamte Anwendung muss in einem Container laufen. |  |

**Szenarien**

|  |  |
| --- | --- |
| **ID** | **Szenario** |
| C1 | Das Unternehmen kann eine eigene Importanwendung schreiben, die die gesammelten Daten aus anderen Anwendungen in diese Anwendung importiert. |
| U1 | Ein neuer Benutzer kann ohne Einführung innerhalb von 10 Minuten mit der Anwendung umgehen und Daten erfassen. |
| M1 | Erweiterungen nach Durchführung des ursprünglichen Projekts können ohne größere Änderungen an der Codebasis gemacht werden. |

## 1.4 Risiken

Da es sich für alle Mitglieder des Projektteams um das erste große Softwareprojekt handelt, gibt es einige Risiken. Die folgenden Risiken haben wir identifiziert und Präventionsmaßnahmen definiert.

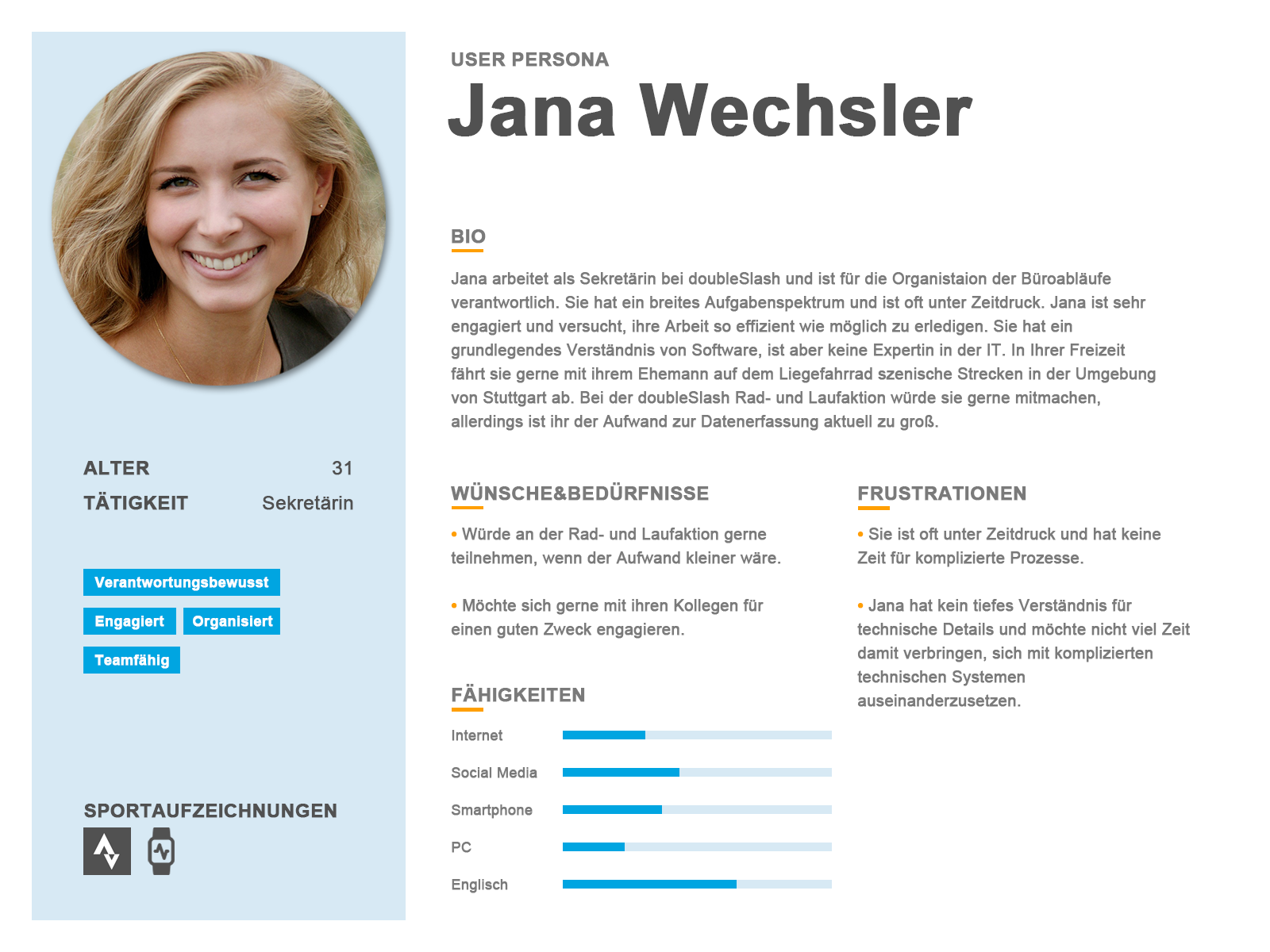
|  |  |
| --- | --- |
| **Risiko** | **Präventionsmaßnahmen** |
| Aufgrund mangelnder Erfahrung könnte die korrekte Zeiteinschätzung für das Umsetzen von User Stories schwierig sein und zu Problemen führen. | * Zusammenarbeit mit dem Betreuer bei den ursprünglichen Schätzungen. * Kontinuierliches Anpassen der Schätzungen, aufgrund von gesammelten Erfahrungen. |
| Ebenso könnte es aufgrund mangelnder Erfahrung mit den verwendeten Technologien und Frameworks zu Schwierigkeiten bei der Umsetzung kommen, was die Qualität des Endprodukts beeinträchtigen würde. | * Workshops mit dem Kunden, bei dem das Projektteam in die neuen Technologien eingeführt wird. * Verwendung von Best Practices. * Regelmäßige Code Reviews im Team, um gegenseitiges Lernen zu fördern. |
| Es könnte zu Problemen kommen, alle geplanten Features bis zum Ende des Projekts vollständig umzusetzen, da die Zeit der Teammitglieder aufgrund anderer akademischer Verpflichtungen begrenzt ist. | * Priorisierung der Features, damit die wichtigsten Funktionen zuerst implementiert werden. * Kontinuierliche Überwachung des Fortschritts, damit Probleme früh identifiziert werden können. * Offene Kommunikation im Team, sodass man sich gegenseitig aushelfen kann. |

## 1.5 Zielgruppe

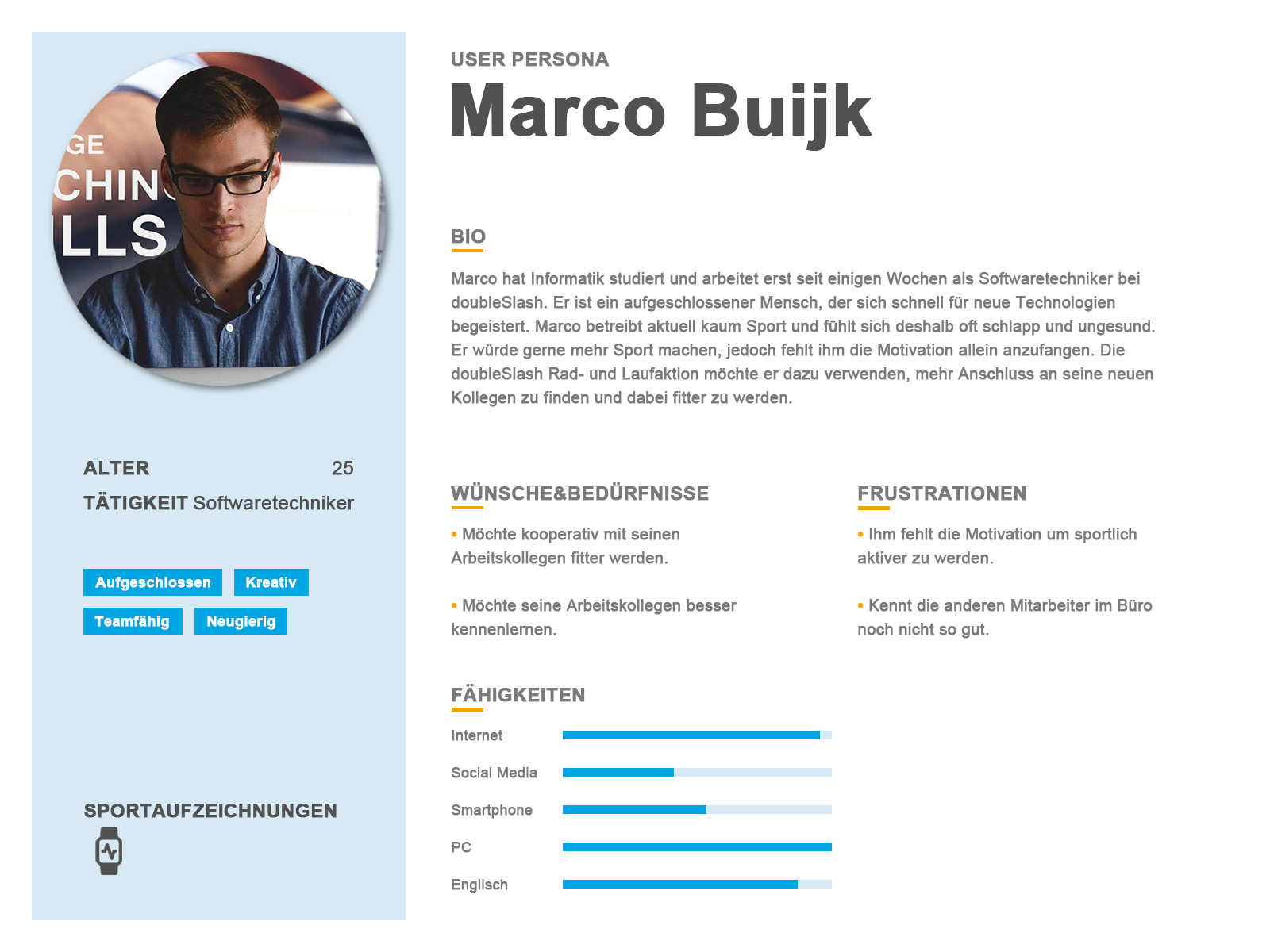
Da das Projekt nur intern verwendet werden soll, sind nur die Mitarbeiter der Firma doubleSlash, die an der firmeninternen Sport Challenge teilnehmen möchten, die Zielgruppe.

## 1.6 Personas

Um eine bessere Vorstellung der zukünftigen Nutzer zu bekommen, haben wir folgende Personas erstellt.







# 2 Projektmanagement

In diesem Abschnitt werden die für das Projekt verwendeten Werkzeuge und Methoden definiert und erläutert.

## 2.1 Agiles Projektmanagement

Eine der Anforderungen an das Projekt ist die Verwendung agiler Methoden für das Projektmanagement. Hier haben wir uns für das Scrum Framework entschieden, da es uns eine gute Zusammenarbeit mit dem Kunden und Flexibilität bei der Umsetzung des Projekts ermöglicht und dennoch durchgeplant ist.

Scrum-Rollenverteilung:

* Product Owner: doubleSlash Net-Business GmbH
* Scrum Master: Jason Patrick Duffy
* Projektteam: Jason Patrick Duffy, Robin Hackh, Tom Nguyen Dinh, Mason Schönherr

### 2.1.1 Terminplan

Wir haben uns für eine Sprintdauer von einer Woche entschieden, da es sich um ein zeitlich kurzes Projekt handelt und auch unsere Treffen mit dem Betreuer und dem Kunden wöchentlich stattfinden.

Der Sprint beginnt wöchentlich donnerstags. Von 10:15 Uhr bis 11:00 Uhr findet das Treffen mit dem Betreuer statt. Danach findet von 11:30 Uhr bis 12:30 Uhr die Sprint Review mit dem Kunden statt. Ab 19:15 Uhr trifft sich das Team für die Retrospektive, woraufhin wir gegen 20 Uhr den nächsten Sprint planen.

Aufgrund des engen Terminplans der Teammitglieder haben wir uns gegen ein tägliches Meeting entschieden, weshalb wir uns innerhalb des Sprints nur zweimal die Woche – sonntags um 18:00 Uhr und dienstags um 13:30 Uhr – treffen.

Zusammengefasst:

* Sprintdauer: 1 Woche
* Sprintbeginn: Donnerstag
* Review: Donnerstag ab 11:30 Uhr
* Retrospektive: Donnerstag ab 19:15 Uhr
* Planung: Donnerstag ab 20:00 Uhr
* Treffen mit dem Betreuer: Donnerstag ab 10:15 Uhr
* Stand-up-Meetings: Sonntag ab 18:00 Uhr und Dienstag ab 13:30 Uhr

Der Abgabetermin für das Projekt ist der 22. Juni 2023.

### 2.1.2 Rollenverteilung

Die Rollenverteilung ist in unserem Projekt sehr flexibel. Die hier genannten Abschnittsleiter sind die Entscheidungsträger bei Entscheidungen in den jeweiligen Abschnitten – beispielsweise haben sie bei Unstimmigkeiten über die Umsetzung eines Features im Team das letzte Wort – und kümmern sich um das Einhalten von Best Practices und den definierten Implementierungsregeln. Die meisten Entscheidungen werden trotzdem versucht im Gesamtteam getroffen zu werden. Zudem bedeutet es nicht, dass der Frontend-Lead nicht auch im Backend arbeiten kann. Die Rollen wurden möglichst nach Menge der Erfahrung im jeweiligen Bereich verteilt.

|  |  |  |
| --- | --- | --- |
| **Person** | **Rollen** | **Rollenbeschreibung** |
| Jason Patrick Duffy | Project Lead  Scrum Master  Backend- & DB-Lead  Full-Stack-Entwickler | Leitet das Projekt und kümmert sich um die Einhaltung der Scrum-Regeln. Ist der Entscheidungsträger bei Entscheidungen im Bezug zum Backend und der Datenbank. Arbeitet an allen Aspekten der Implementierung mit. |
| Robin Hackh | Frontend- & UX-Lead  Full-Stack-Entwickler | Trifft Entscheidungen über das Frontend und der Benutzererfahrung. Arbeitet an allen Aspekten der Implementierung mit. |
| Mason Schönherr | Documentation-Lead  Teamsprecher  Full-Stack-Entwickler | Kümmert sich darum, dass Implementierungen ausreichend dokumentiert werden. Spricht in der Öffentlichkeit für das Team. Arbeitet an allen Aspekten der Implementierung mit. |
| Tom Nguyen Dinh | QA-Lead  Full-Stack-Entwickler | Prüft, dass ausreichend und richtig getestet wird. Arbeitet an allen Aspekten der Implementierung mit. |

### 2.1.3 Definition of Ready

Damit eine User Story in den Sprint aufgenommen werden kann, müssen folgende Kriterien erfüllt sein:

* Die User Story ist für alle Beteiligten klar, verständlich und vollständig beschrieben.
* Der Aufwand für die User Story wurde vom Projektteam geschätzt.
* Alle sicherheits- und datenschutzrelevanten Aspekte wurden geprüft und berücksichtigt.
* Es wurde geprüft, dass die User Story innerhalb der Architektur umsetzbar ist.
* Es wurden klare Akzeptanzkriterien formuliert, die von allen Beteiligten verstanden werden und testbar sind.

### 2.1.4 Definition of Done

Eine User Story gilt als fertiggestellt, wenn folgende Kriterien erfüllt sind:

* Alle Akzeptanzkriterien der User Story sind erfüllt.
* Die für die Umsetzung benötigte Zeit wurde eingetragen.
* Eine Code Review wurde durchgeführt
* Die Branch wurde mit der Sprint-Branch zusammengefügt.
* Implementierungsregeln wurden eingehalten.
* Der Code steht in der Versionsverwaltung zur Verfügung.
* Die Implementierung der User Story hat keine kritischen Fehler.
* Alle Tests waren erfolgreich und folgen der Testing Policy.

## 2.2 Projektverwaltung

Zur Projektverwaltung verwenden wir Atlassian Jira. Dies ermöglicht uns an einem Ort unsere Epics und User Stories festzuhalten sowie unsere Sprints zu planen. Zudem erhalten wir so einen schnellen Überblick über den aktuellen Fortschritt und können Probleme schneller identifizieren.

Link zum Jira-Board: <https://jduffy.atlassian.net/jira/software/projects/SCP/boards/2>

### 2.2.1 User Story Regeln

Beim Erstellen von User Stories sollte darauf geachtet werden, immer eine Priorität anzugeben.

Die Prioritäten haben folgende Bedeutung:

* Höchste: Grundlegende Funktion. Muss implementiert werden.
* Hoch: Wichtige Funktion. Sollte auf jeden Fall implementiert werden.
* Mittel: Geforderte Funktion. Sollte implementiert werden.
* Niedrig: Nicht geforderte Funktion. Kann implementiert werden.
* Niedrigste: Nicht geforderte Funktion. Sollte nur bei Zeitüberschuss implementiert werden.

## 2.3 Versionsverwaltung

Zur Versionsverwaltung verwenden wir das interne GitLab der Hochschule Esslingen.

Wir haben uns für GitLab entschieden, da wir dort einen Code Runner zur Verfügung haben, der uns das Umsetzen von CI/CD ermöglicht, was mit GitHub nicht möglich gewesen wäre. Zudem benutzen wir das hochschulinterne GitLab, da wir ansonsten auf die Begrenzung der CI/CD Minuten achten müssten.

Link zum GitLab (nur aus dem HS Esslingen-VPN erreichbar):   
<https://gitlab.hs-esslingen.de/jaduit00/Sport-Challenge-Projekt>

### 2.3.1 Commit Regeln

Um Commits einheitlich und übersichtlich zu halten, verwenden wir die Conventional Commits-Spezifikation[[2]](#footnote-3) mit einem kleinen Zusatz. Commits sollten immer nur einem Typ zugehörig sein. Commits sollten also regelmäßig gemacht werden, um übergroße Commits, der mehrere Typen angehören, zu verhindern.

**Aufbau eines Conventional Commits:**  
<Typ>: Beschreibung (Titel des Commits)  
Optionaler Textkörper, der den Commit genauer beschreibt.  
Refs: SCP-0

Time: X h

Folgende Typen verwenden wir:

|  |  |
| --- | --- |
| **Typ** | **Beschreibung** |
| fix | Ein Fehler in der Codebasis wird behoben. |
| feat | Eine neue Funktion wurde in die Codebasis eingeführt. |
| ci | Änderungen an der CI/CD Pipeline. |
| docs | Ein Dokument wurde verändert oder hinzugefügt. |
| test | Ein Test wurde verändert oder hinzugefügt. |
| revert | Ein Commit wurde rückgängig gemacht. |

In der Fußzeile verwenden wir drei Typen:

|  |  |
| --- | --- |
| **Typ** | **Beschreibung** |
| BREAKING CHANGE | Es werden tiefgründige Änderungen in der API eingeführt.  Immer in Kombination mit einem ! beim Typ. |
| Refs: SCP-0, SCP-1, … | In Jira referenzierte Tickets, die zum Commit gehören. |
| Time: X h | Die für die Umsetzung des Commits benötigte Zeit in Stunden.  Zur Verbesserung der Lesbarkeit, ist die kleinste mögliche Angabe „0.1 h“ und kann von dort in 0.1-Schritten erhöht werden.  Dies ist nicht Teil der Conventional Commits Spezifikation. |

**Beispiel**

feat!: Neues System zur Verwaltung von Nutzerdaten implementiert  
Nutzerdaten werden jetzt über das neue System verwaltet. Durch effizientere Nutzung der Datenbank, ist die Reaktionsgeschwindigkeit der Anwendung nun deutlich schneller.  
BREAKING CHANGE: Alte Nutzerdaten sind mit der neuen Version inkompatibel.  
Refs: SCP-12  
Time: 4.7 h

### 2.3.2 Branch Regeln

Ein Bild, das Diagramm enthält.

Automatisch generierte Beschreibung

Abbildung 1: Beispiel für Branches innerhalb eines einzelnen Sprints

Wir verwenden eine primäre Branch namens *Main*.

Am Anfang eines Sprints wird von Main eine Branch abgeleitet, die *Sprint X* heißt. *X* steht hierfür für die aktuelle Sprintnummer, die man in Jira einsehen kann.

Innerhalb des Sprints werden für das Implementieren von Features von der Sprint-Branch Feature-Branches abgeleitet. Diese Branches werden *Feature X* genannt, wobei *X* für das spezifische Feature steht.

Ein Bild, das Diagramm enthält.

Automatisch generierte Beschreibung

Abbildung 2: Ausgefülltes Beispiel für die Verwendung von Branches

Wenn das Feature implementiert ist, wird ein Merge Request durchgeführt, der von mindestens einer weiteren Person überprüft werden **muss**. Am Ende des Sprints wird die Sprint-Branch wieder mit der Main-Branch zusammengeführt, sodass nur noch die Main-Branch existiert.

## 2.4 Teamkommunikation

Die teaminterne Kommunikation findet über eine Discord Gruppe und persönliche Treffen statt.

## 2.5 Implementierung

Um die Lesbarkeit der Implementierung zu garantieren, haben wir einige Regeln definiert.

|  |  |  |
| --- | --- | --- |
| **Regel** | **Beschreibung** | **Beispiel** |
| Code ausreichend kommentieren | Code, der nicht selbsterklärend ist, sollte immer eine kurze Beschreibung, sowie den Autor beinhalten. Wenn möglich sollten sprachenspezifische Features, wie JavaDoc, eingebunden werden.  Selbsterklärender Code sind beispielsweise simple Getter und Setter. | Ein Bild, das Text enthält.  Automatisch generierte Beschreibung  Abbildung 3: Beispiel in Java mit JavaDoc |
| Einheitlichen Einrückungsstil verwenden | Zur besseren Lesbarkeit sollten entweder der **Allman** oder **Kernighan & Ritchie** Einrückungsstil verwendet werden. | Ein Bild, das Diagramm, Text, Brief enthält.  Automatisch generierte Beschreibung  Abbildung 4: Verschiedene Einrückungsstile [1] |

### 2.5.1 Testing Policy

Die Testing Policy hat das Ziel, sicherzustellen, dass die Implementierung richtig und fehlerfrei implementiert ist. Unser Ziel ist es, eine Testabdeckung von 80% zu erreichen.

Für das Frontend verwenden wir die *Jest* Testing-Bibliothek, da es bereits in React integriert ist und empfohlen wird [2]. Außerdem beinhaltet Jest alle für das Projekt benötigten Funktionen.

Beim Backend verwenden wir die *Spring Boot Starter Test* Abhängigkeit. Sie beinhaltet bereits alle für uns nötigen Funktionen und ist sehr gut mit Spring Boot integriert.

Die Tests sollten dabei möglichst mittels Verwendung von CI/CD automatisiert werden.

# 3 Funktionsumfang

In diesem Abschnitt werden unsere User Stories aufgelistet und mit der in [Abschnitt 2.2.1](#_2.2.1_User_Story) angesprochenen Priorität versehen. Die Ticket ID wird dabei automatisch von Jira generiert.

Die User Stories werden in folgende Epics eingeteilt:

|  |  |
| --- | --- |
| **Epic** | **Beschreibung** |
| Benutzerverwaltung | User Stories, die hauptsächlich mit der Verwaltung der Benutzer zusammenhängen. Beispiele hierfür sind Login oder Nutzerdaten bearbeiten. |
| Challengeverwaltung | User Stories, die sich hauptsächlich um die Verwaltung der Challenges drehen. Beispiele hierfür sind das Erstellen und Bearbeiten von Challenges. |
| Sportverwaltung | User Stories, die mit dem Verwalten von Sportarten zu tun haben. Beispiele sind das Erstellen einer neuen oder das Bearbeiten einer vorhandenen Sportart. |
| Teamverwaltung | User Stories, die das Verwalten von Teams definieren. Beispiele sind das Erstellen eines Teams oder das Hinzufügen von Mitgliedern. |

## 3.1 User Stories

Hier werden alle von uns erstellten User Stories zusammen mit ihrer Ticket ID, ihrer Epic-Zuweisung, dem Autor, der Priorität sowie der vorläufigen Zeitschätzung aufgelistet.

|  |  |
| --- | --- |
| Ticket ID | SCP-2 |
| Epic | Benutzerverwaltung |
| User Story | Um die Website verwenden zu können, möchte ich mich als Benutzer mit meinem doubleSlash Konto einloggen können. |
| Erstellt von | Jason Patrick Duffy |
| Priorität | Höchste |
| Zeitschätzung | 1 Tag 2 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-5 |
| Epic | Challengeverwaltung |
| User Story | Um einen Überblick über alle laufenden Challenges zu haben, möchte ich als Benutzer eine Übersicht mit allen aktiven Challenges und einer jeweiligen Kurzinfo dazu. |
| Erstellt von | Jason Patrick Duffy |
| Priorität | Höchste |
| Zeitschätzung | 6 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-7 |
| Epic | Sportverwaltung |
| User Story | Als Administrator möchte ich in der Lage sein, eine neue Sportart hinzuzufügen. |
| Erstellt von | Mason Schönherr |
| Priorität | Niedrig |
| Zeitschätzung | 4 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-10 |
| Epic | Teamverwaltung |
| User Story | Um an einer Challenge teilzunehmen, möchte ich als Benutzer einem Team beitreten können. |
| Erstellt von | Jason Patrick Duffy |
| Priorität | Hoch |
| Zeitschätzung | 2 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-13 |
| Epic | Challengeverwaltung |
| User Story | Als Benutzer will ich eine Challenge mit den gegebenen Spezifikationen anlegen können. |
| Erstellt von | Robin Hackh |
| Priorität | Höchste |
| Zeitschätzung | 1 Tag |

|  |  |
| --- | --- |
| Ticket ID | SCP-14 |
| Epic | Challengeverwaltung |
| User Story | Als Benutzer möchte ich die Spezifikationen wie z.B. die Sportarten einer bereits existierenden Challenge ändern können. |
| Erstellt von | Robin Hackh |
| Priorität | Mittel |
| Zeitschätzung | 6 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-15 |
| Epic | Sportverwaltung |
| User Story | Um neue Anregung für die Challenge zu schaffen, möchte ich als Benutzer mehrere/einzelne temporäre Bonusaktionen für die Challenge erstellen können. |
| Erstellt von | Robin Hackh |
| Priorität | Hoch |
| Zeitschätzung | 1 Tag 4 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-16 |
| Epic | Sportverwaltung |
| User Story | Um eine Challenge fairer zu gestalten, möchte ich als Administrator für Sportarten einen Multiplikator festlegen können. |
| Erstellt von | Robin Hackh |
| Priorität | Mittel |
| Zeitschätzung | 2 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-17 |
| Epic | Teamverwaltung |
| User Story | Um an einer Challenge teilzunehmen, möchte ich als Benutzer ein Team erstellen können, das einer aktiven Challenge zugeordnet wird. |
| Erstellt von | Robin Hackh |
| Priorität | Hoch |
| Zeitschätzung | 6 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-18 |
| Epic | Challengeverwaltung |
| User Story | Um keine Bonusaktionen zu verpassen, möchte ich als Benutzer über anstehende/laufende Boni per Mail informiert werden. |
| Erstellt von | Robin Hackh |
| Priorität | Mittel |
| Zeitschätzung | 1 Tag 4 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-19 |
| Epic | Challengeverwaltung |
| User Story | Um nicht von der Sport Challenge abgelenkt zu werden, möchte ich als Benutzer die Benachrichtigungsmails abmelden können. |
| Erstellt von | Jason Patrick Duffy |
| Priorität | Niedrig |
| Zeitschätzung | 2 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-20 |
| Epic | Challengeverwaltung |
| User Story | Um einen Fortschritt in der Challenge zu erzielen, möchte ich als Benutzer meine zurückgelegten Kilometer in einer Challenge, in der ich teilnehme, eintragen können. |
| Erstellt von | Robin Hackh |
| Priorität | Höchste |
| Zeitschätzung | 5 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-21 |
| Epic | Challengeverwaltung |
| User Story | Um herauszufinden wie mein Team und ich bei der Challenge abschneiden, möchte ich als Benutzer eine mit Charts aufbereitete Seite der Challenges. |
| Erstellt von | Robin Hackh |
| Priorität | Mittel |
| Zeitschätzung | 3 Tage |

|  |  |
| --- | --- |
| Ticket ID | SCP-22 |
| Epic | Challengeverwaltung |
| User Story | Um meine Kilometer nicht jedes mal von Hand eingeben zu müssen, möchte ich als Benutzer diese mit einem Bild meines Tachos eintragen können. |
| Erstellt von | Robin Hackh |
| Priorität | Niedrigste |
| Zeitschätzung | 4 Tage |

|  |  |
| --- | --- |
| Ticket ID | SCP-23 |
| Epic | Benutzerverwaltung |
| User Story | Um mich von anderen Benutzern abzuheben, möchte ich als Benutzer mein Profilbild ändern können. |
| Erstellt von | Robin Hackh |
| Priorität | Niedrig |
| Zeitschätzung | 4 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-24 |
| Epic | Challengeverwaltung |
| User Story | Um Ordnung bei den Challenges zu waren, möchte ich als Administrator, dass nur ausgewählte Benutzer berechtigt sind Spezifikationen der Challenge zu ändern. |
| Erstellt von | Robin Hackh |
| Priorität | Niedrigste |
| Zeitschätzung | 6 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-25 |
| Epic | Benutzerverwaltung |
| User Story | Um mein Profil ansprechender zu gestalten, möchte ich als Benutzer meinem Profil ein Motto hinzufügen können. |
| Erstellt von | Robin Hackh |
| Priorität | Niedrigste |
| Zeitschätzung | 2 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-26 |
| Epic | Benutzerverwaltung |
| User Story | Um mein Profil auf dem aktuellsten Stand zu halten, möchte ich als Benutzer meine Profildaten ändern können. |
| Erstellt von | Robin Hackh |
| Priorität | Mittel |
| Zeitschätzung | 5 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-28 |
| Epic | Sportverwaltung |
| User Story | Um Fehler zu korrigieren, möchte ich als Administrator die Daten einer Sportart bearbeiten können. |
| Erstellt von | Jason Patrick Duffy |
| Priorität | Mittel |
| Zeitschätzung | 4 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-29 |
| Epic | Challengeverwaltung |
| User Story | Um die doppelte Datenerfassung zu ersparen, möchte ich als Benutzer meine Kilometerdaten aus der Strava-App importieren können. |
| Erstellt von | Jason Patrick Duffy |
| Priorität | Niedrig |
| Zeitschätzung | 1 Tag |

|  |  |
| --- | --- |
| Ticket ID | SCP-30 |
| Epic | Challengeverwaltung |
| User Story | Um die Anwendung in Zukunft erweitern zu können, möchte ich als Entwickler bei doubleSlash eine API zum Anbinden weiterer Apps haben. |
| Erstellt von | Jason Patrick Duffy |
| Priorität | Niedrigste |
| Zeitschätzung | 1 Tag 6 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-31 |
| Epic | Benutzerverwaltung |
| User Story | Um mich von allem abzumelden, möchte ich als Benutzer mein Konto vollständig löschen können. |
| Erstellt von | Jason Patrick Duffy |
| Priorität | Niedrig |
| Zeitschätzung | 4 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-32 |
| Epic | Benutzerverwaltung |
| User Story | Um ungewollte Änderungen zu vermeiden, möchte ich als Administrator Benutzergruppen erstellen und ihnen bestimmte Rechte zuweisen können. |
| Erstellt von | Jason Patrick Duffy |
| Priorität | Niedrig |
| Zeitschätzung | 2 Tage 6 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-33 |
| Epic | Benutzerverwaltung |
| User Story | Um die Anwendung verwenden zu können, möchte ich als Benutzer meine bevorzugte Sprache auswählen können. |
| Erstellt von | Jason Patrick Duffy |
| Priorität | Niedrig |
| Zeitschätzung | 6 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-34 |
| Epic | Benutzerverwaltung |
| User Story | Um nachvollziehen zu können, wer welche Änderungen gemacht hat, möchte ich als Administrator eine Übersicht über alle gemachten Änderungen und wer diese getätigt hat haben. |
| Erstellt von | Jason Patrick Duffy |
| Priorität | Niedrigste |
| Zeitschätzung | 1 Tag 4 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-35 |
| Epic | Teamverwaltung |
| User Story | Um die Challenge zu organisieren, möchte ich als Administrator Benutzer aus Teams entfernen können. |
| Erstellt von | Jason Patrick Duffy |
| Priorität | Niedrig |
| Zeitschätzung | 4 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-36 |
| Epic | Teamverwaltung |
| User Story | Um meinen Kollegen einfach in mein Team zu bekommen, möchte ich als Benutzer andere Benutzer in eine Challenge und mein Team einladen können. |
| Erstellt von | Jason Patrick Duffy |
| Priorität | Niedrig |
| Zeitschätzung | 1 Tag |

|  |  |
| --- | --- |
| Ticket ID | SCP-37 |
| Epic | Teamverwaltung |
| User Story | Um Fehler zu korrigieren, möchte ich als Benutzer die Daten meines Teams ändern können. |
| Erstellt von | Jason Patrick Duffy |
| Priorität | Mittel |
| Zeitschätzung | 5 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-38 |
| Epic | Challengeverwaltung |
| User Story | Um die Daten der Challenge weiterverarbeiten zu können, möchte ich als Administrator die Daten der Challenge exportieren können. |
| Erstellt von | Jason Patrick Duffy |
| Priorität | Niedrigste |
| Zeitschätzung | 1 Tag |

|  |  |
| --- | --- |
| Ticket ID | SCP-40 |
| Epic | Teamverwaltung |
| User Story | Um eine faire Challenge zu gestalten, möchte ich als Administrator, dass die Kilometer jedes Teams als Durchschnitt der Mitglieder gerechnet werden. |
| Erstellt von | Tom Nguyen Dinh |
| Priorität | Mittel |
| Zeitschätzung | 1 Stunde 30 Minuten |

|  |  |
| --- | --- |
| Ticket ID | SCP-46 |
| Epic | Teamverwaltung |
| User Story | Um einen besseren Teamgeist zu fördern, möchte ich als Benutzer dem Team gerne einem Namen geben können. |
| Erstellt von | Tom Nguyen Dinh |
| Priorität | Niedrig |
| Zeitschätzung | 2 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-58 |
| Epic | Teamverwaltung |
| User Story | Um eine dynamischere Erfahrung zu haben, möchte ich als Benutzer für jede Challenge ein anderes Team auswählen können. |
| Erstellt von | Tom Nguyen Dinh |
| Priorität | Mittel |
| Zeitschätzung | 0 Minuten |

|  |  |
| --- | --- |
| Ticket ID | SCP-60 |
| Epic | Teamverwaltung |
| User Story | Um mir später Arbeit zu ersparen, möchte ich als Benutzer mich bereits vor Challengebeginn zur Challenge anmelden können. |
| Erstellt von | Tom Nguyen Dinh |
| Priorität | Niedrig |
| Zeitschätzung | 2 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-61 |
| Epic | Teamverwaltung |
| User Story | Um eine faire Challenge zwischen den Teams zu haben, möchte ich als Administrator, dass Teilnehmer ihr Team nicht während einer laufenden Challenge wechseln können. |
| Erstellt von | Tom Nguyen Dinh |
| Priorität | Niedrig |
| Zeitschätzung | 0 Minuten |

|  |  |
| --- | --- |
| Ticket ID | SCP-62 |
| Epic | Sportverwaltung |
| User Story | Bei Veränderungen, wie das neue Hinzukommen einer Sportart, möchte ich als Benutzer eine E-Mail erhalten. |
| Erstellt von | Tom Nguyen Dinh |
| Priorität | Niedrig |
| Zeitschätzung | 2 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-98 |
| Epic | Challengeverwaltung |
| User Story | Als User möchte ich in der Lage sein aus einer Challenge auszusteigen, falls ich bemerke, dass sie doch nichts für mich ist. |
| Erstellt von | Mason Schönherr |
| Priorität | Niedrigste |
| Zeitschätzung | 4 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-99 |
| Epic | Challengeverwaltung |
| User Story | Als aktiver User möchte in der Lage sein meine Aktivitäten auf mehreren Seiten der Website eintragen zu können, um beim Hin- und Herklicken Zeit zu sparen. |
| Erstellt von | Mason Schönherr |
| Priorität | Niedrig |
| Zeitschätzung | 1 Tag |

|  |  |
| --- | --- |
| Ticket ID | SCP-100 |
| Epic | Challengeverwaltung |
| User Story | Als User möchte ich eine graphische Historie meiner Aktivitäten sehen können. |
| Erstellt von | Mason Schönherr |
| Priorität | Niedrig |
| Zeitschätzung | 1 Tag |

|  |  |
| --- | --- |
| Ticket ID | SCP-103 |
| Epic | Benutzerverwaltung |
| User Story | Um weniger Probleme bei der Nutzung zu haben, möchte ich als Benutzer meine bevorzugten Einheiten (Kilometer, Meilen, Yards, ...) auswählen können. |
| Erstellt von | Jason Patrick Duffy |
| Priorität | Niedrigste |
| Zeitschätzung | 6 Stunden |

|  |  |
| --- | --- |
| Ticket ID | SCP-109 |
| Epic | Challengeverwaltung |
| User Story | Um Fehler zu korrigieren, möchte ich als Benutzer in der Lage sein, meine gutgeschriebenen Kilometer zu widerrufen oder zu ändern. |
| Erstellt von | Mason Schönherr |
| Priorität | Mittel |
| Zeitschätzung |  |

|  |  |
| --- | --- |
| Ticket ID | SCP-110 |
| Epic | Sportverwaltung |
| User Story | Um die Challenge weiter personalisieren zu können, möchte ich als Administrator für jede Sportart einen eigenen Multiplikator auswählen können. |
| Erstellt von | Mason Schönherr |
| Priorität | Hoch |
| Zeitschätzung | 3 Stunden |

# 4 UI-Entwürfe

In diesem Abschnitt wird unser Prototyp vorgestellt. Außerdem findet sich hier eine Tabelle der vom Kunden verwendeten Farben.

**Farbcodes**

|  |  |
| --- | --- |
| **Farbe** | **Farbcode** |
|  | Markenfarbe: #00A5E1 |
|  | Sekundär Unternehmensfarbe: #D7E9F4 |
|  | Sekundär Orange: #ff9f00 |
|  | Schrift Dunkelgrau: #515151 |
|  | Sekundär Dunkelgrau: #7A7A7A |
|  | Sekundär Hellgrau: #C6C6C6 |
|  | Akzentfarbe: #32547B |
|  | Akzentfarbe: #B9D478 |
|  | Akzentfarbe: #8B593E |
|  | Akzentfarbe: #C63328 |

# 5 Architektur

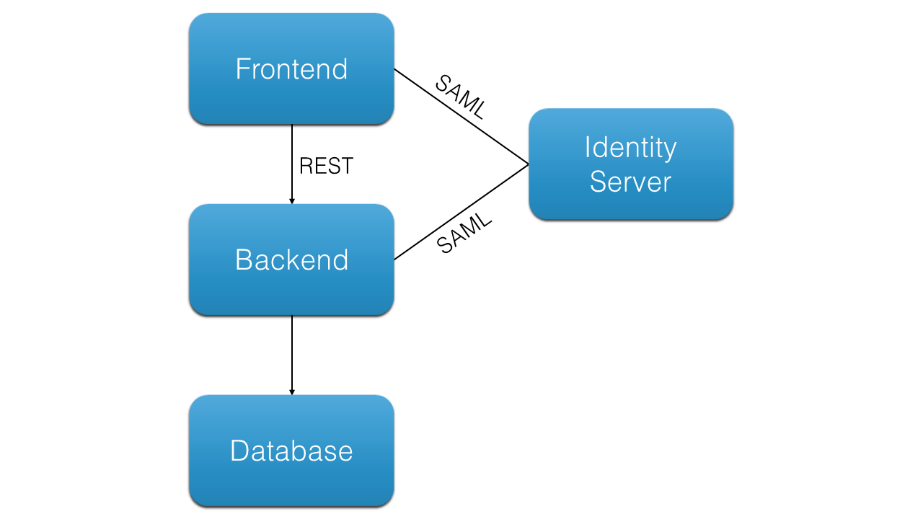


Abbildung 5: Grundlegende Architektur

Bei der Wahl der Architektur war es besonders wichtig, dass die Teammitglieder bereits grundlegende Kenntnisse in den zugrundeliegenden Programmiersprachen haben und die kommerzielle Nutzung der Frameworks erlaubt ist. Wenn möglich wollten wir zudem Open Source Frameworks verwenden.

Für das Frontend haben wir uns daher auf das React Framework geeinigt, da jedes Teammitglied bereits Erfahrung bei der Arbeit mit JavaScript hat und die hohe Verbreitung des Frameworks uns bei Fragen und Problemen helfen sollte.

Beim Backend fiel die Wahl auf das Spring Boot Framework, da das Team ebenfalls bereits Erfahrung bei der Entwicklung mit Java hat und auch hier der hohe Grad der Verbreitung das Problemlösen erleichtern sollte. Das Spring Boot Framework eignet sich zudem ideal zum Aufbau einer REST-Schnittstelle.

Da nur Mitarbeiter des Kunden Zugriff auf das System haben dürfen, stellt die Authentifizierung von Benutzern einen wichtigen Punkt dar. Hier haben wir uns für Keycloak entschieden, da es sich sehr gut mit der von den Kunden verwendeten SAML-Schnittstelle verbinden lässt.

Schließlich entschieden wir uns beim Datenbanksystem für PostgreSQL. Auch hier sollte der hohe Verbreitungsgrad bei der Problemlösung helfen. Zudem ermöglicht die Funktionsvielfalt von PostgreSQL die spätere Erweiterung der Anwendung mit komplexeren Aufgabenstellungen – es ist also zukunftsorientiert.

Daraus ergibt sich folgende Architektur:

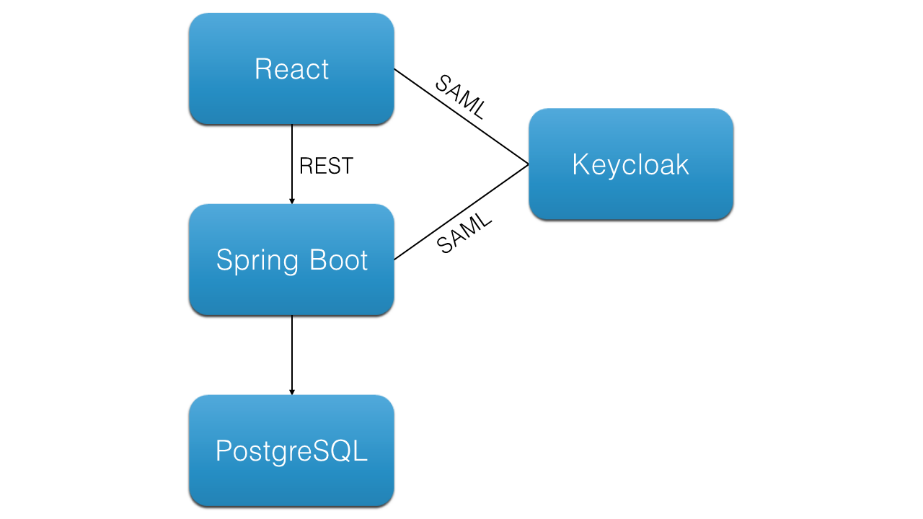


Abbildung 6: Architektur mit den von uns gewählten Frameworks

## 5.1 Lizenzen

In diesem Abschnitt werden die Lizenzen der verwendeten Frameworks und des Projekts aufgelistet und beschrieben.

### 5.1.1 Projekt

Für unser Projekt haben wir uns für die MIT Lizenz [3] entschieden. Die Lizenz erlaubt die kommerzielle und private Nutzung und das Modifizieren und Verteilen der Software. Es wird keine Garantie oder Haftung angeboten. Eine Kopie der Lizenz muss mitgeliefert werden.

Mit der MIT Lizenz kann der Kunde in Zukunft das Projekt frei weiterentwickeln und erweitern, was eine der Anforderungen an das Projekt ist.

### 5.1.2 React

React wird ebenfalls mit der MIT Lizenz ausgeliefert [4]. Die Lizenz ist identisch zu unserem Projekt. Die Lizenz erlaubt die kommerzielle und private Nutzung und das Modifizieren und Verteilen der Software. Es wird keine Garantie oder Haftung angeboten. Eine Kopie der Lizenz muss mitgeliefert werden.

### 5.1.3 Java Spring Boot

Java Spring Boot verwendet die Apache Lizenz 2.0 [5]. Die Lizenz erlaubt die kommerzielle und private Nutzung, das Modifizieren und Verteilen der Software sowie die Verwendung in Patenten. Nicht erlaubt ist die Verwendung von Apache Trademarks. Zudem wird keine Garantie oder Haftung angeboten. Software, die Anwendungen mit der Apache Lizenz 2.0 verwendet, muss eine Kopie der Lizenz beinhalten. Änderungen an der lizenzierten Software müssen klar markiert sein. Hat die lizenzierte Software eine „NOTICE“-Textdatei, muss diese ebenfalls mitgeliefert werden.

### 5.1.4 Keycloak

Bei Keycloak wird ebenfalls die Apache Lizenz 2.0 verwendet [6]. Der Lizenztext ist identisch zu dem von Java Spring Boot. Die Lizenz erlaubt die kommerzielle und private Nutzung, das Modifizieren und Verteilen der Software sowie die Verwendung in Patenten. Nicht erlaubt ist die Verwendung von Apache Trademarks. Zudem wird keine Garantie oder Haftung angeboten. Software, die Anwendungen mit der Apache Lizenz 2.0 verwendet, muss eine Kopie der Lizenz beinhalten. Änderungen an der lizenzierten Software müssen klar markiert sein. Hat die lizenzierte Software eine „NOTICE“-Textdatei, muss diese ebenfalls mitgeliefert werden.

### 5.1.5 PostgreSQL

PostgreSQL wird unter der PostgreSQL Lizenz veröffentlicht [7]. Die Lizenz erlaubt das Verwenden, Kopieren, Modifizieren und Verteilen der Software für jeglichen Verwendungszweck ohne Gebühr oder Einwilligung der Entwickler, solange der Urheberrechtshinweis mitverteilt wird.

## 5.2 Diagramme

In diesem Abschnitt befinden sich einige Diagramme der Architektur und Implementierung.

### 5.2.1 ER-Diagramm

Folgendes ER-Diagramm haben wir für die Datenbankstruktur erstellt.

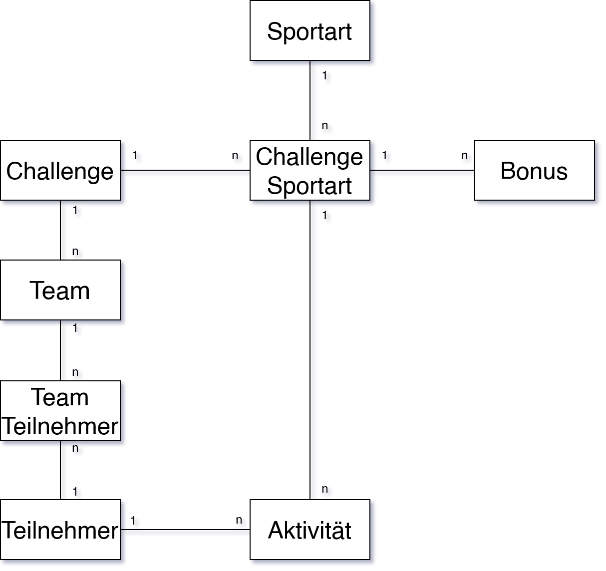


Abbildung : ER-Diagramm der Datenbankstruktur

### 5.2.2 Klassendiagramm

Folgendes Klassendiagramm haben wir für die Implementierung erstellt.

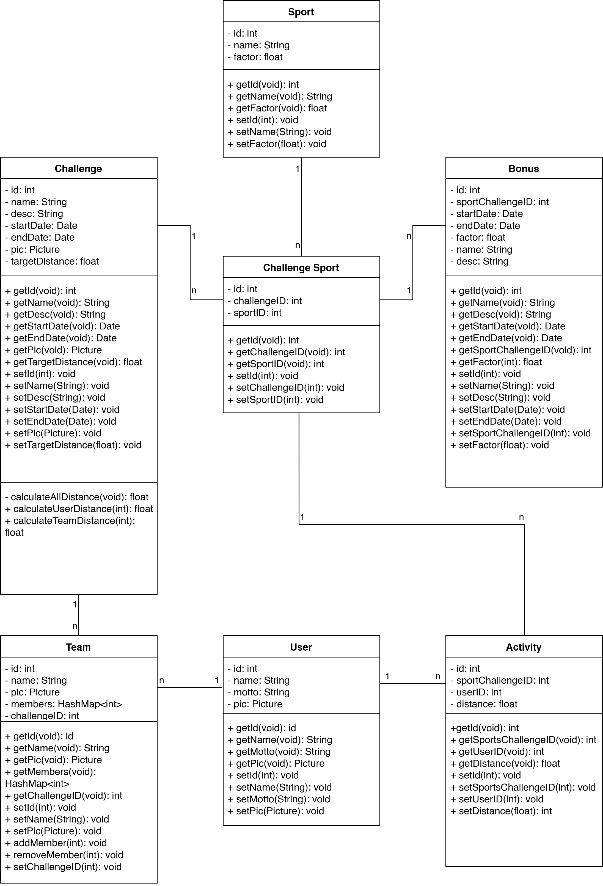


Abbildung : Vorläufiges Klassendiagramm der Implementierung

### 5.2.3 Ablaufdiagramm

Wir haben den Nutzungsablauf einer einzelnen Challenge modelliert und dabei folgendes Ergebnis erhalten.

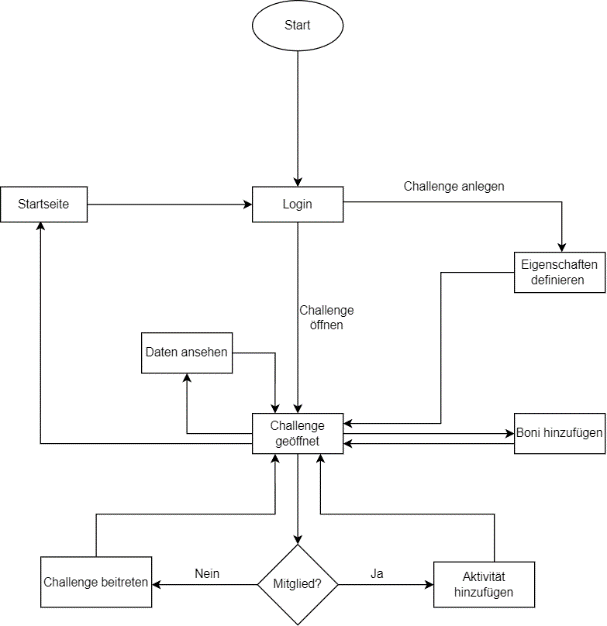


Abbildung : Ablaufdiagramm einer einzelnen Challenge

# 6 Aufwandschätzung

Für die Sprintplanung haben wir darauf geachtet, dass jeder Sprint eine Arbeitszeit von etwa vier Tagen aufweist. Das entspricht pro Woche und pro Teammitglied etwa einem Arbeitstag, also acht Stunden.

So stellen wir sicher, dass wir etwas Überschuss bei der Umsetzung haben, falls wir auf unvorhergesehene Probleme stoßen sollten.

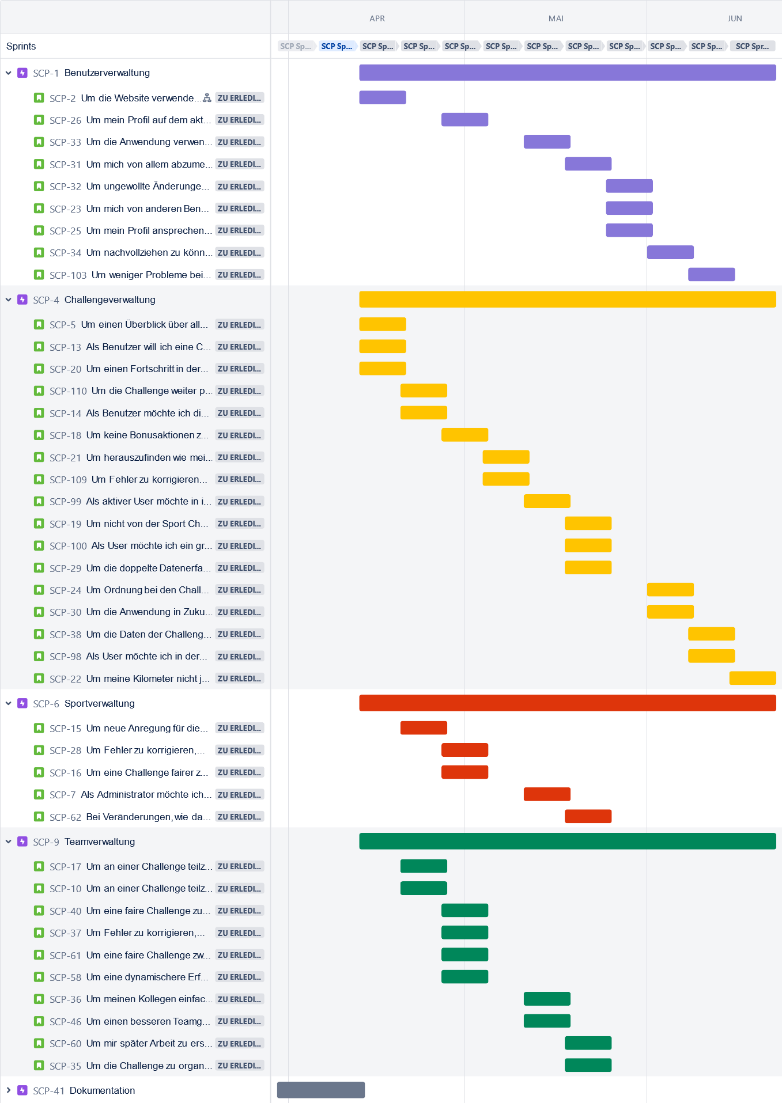


Abbildung 10: Vorläufige Sprintplanung zum Stand 12. April 2023

Rechnet man die Zeitschätzungen der [User Stories](#_3.1_User_Stories) zusammen, kommen wir so auf einen Arbeitsaufwand von *290.5* Stunden, was einer wöchentlichen Arbeitszeit von etwa *29* Stunden und *7* Stunden pro Person entspricht.

Rechnet man dazu noch wöchentlich folgende Punkte pro Teammitglied:

* Treffen mit dem Betreuer: *0.75* Stunden
* Sprint Review: *1* Stunde
* Sprint Retrospektive: *0.75* Stunden
* Sprint Planung: *1* Stunde
* Stand-up-Meetings: *2 \* 0.5* Stunden

Ergibt sich über zehn Wochen ein wöchentlicher Arbeitsaufwand von etwa *12* Stunden pro Teammitglied.

Es ist allerdings wichtig zu beachten, dass dieser Zeitplan die Umsetzung aller User Stories beinhaltet. Also auch die, die nicht direkt Teil der Kundenforderungen sind. Rechnet man nur mit den explizit geforderten Funktionen, kommt man auf eine Arbeitszeit von *98.5* Stunden, was einem wöchentlichen Arbeitsaufwand von etwa *7* Stunden pro Person entspricht.

# Quellenverzeichnis

|  |  |
| --- | --- |
| [1] | Fermat's Library, „Darstellung von Einrückungsstilen,“ 10 Oktober 2022. [Online]. Available: https://twitter.com/fermatslibrary/status/1582719971146309632/. [Zugriff am 4 April 2023]. |
| [2] | Meta, „React Testing Environments,“ [Online]. Available: https://legacy.reactjs.org/docs/testing-environments.html. [Zugriff am 10 April 2023]. |
| [3] | „MIT Lizenz,“ [Online]. Available: https://opensource.org/license/mit/. [Zugriff am 6 April 2023]. |
| [4] | Meta, „React Lizenz,“ [Online]. Available: https://github.com/facebook/react/blob/main/LICENSE. [Zugriff am 6 April 2023]. |
| [5] | Spring Boot, „Lizenz,“ [Online]. Available: https://github.com/spring-projects/spring-boot/blob/main/LICENSE.txt. [Zugriff am 6 April 2023]. |
| [6] | Keybloack, „Lizenz,“ [Online]. Available: https://github.com/keycloak/keycloak-community/blob/main/LICENSE.txt. [Zugriff am 6 April 2023]. |
| [7] | PostgreSQL, „Lizenz,“ [Online]. Available: https://www.postgresql.org/about/licence/. [Zugriff am 6 April 2023]. |
| [8] | Conventional Commits, „Spezifikation,“ [Online]. Available: https://www.conventionalcommits.org/de/v1.0.0/. [Zugriff am 4 April 2023]. |
| [9] | ISO, „ISO25010,“ [Online]. Available: https://iso25000.com/index.php/en/iso-25000-standards/iso-25010/. [Zugriff am 4 April 2023]. |

# Abbildungsverzeichnis

[Abbildung 1: Beispiel für Branches innerhalb eines einzelnen Sprints 10](#_Toc132226803)

[Abbildung 2: Ausgefülltes Beispiel für die Verwendung von Branches 11](#_Toc132226804)

[Abbildung 3: Beispiel in Java mit JavaDoc 11](#_Toc132226805)

[Abbildung 4: Verschiedene Einrückungsstile [1] 11](#_Toc132226806)

[Abbildung 5: Grundlegende Architektur 22](#_Toc132226807)

[Abbildung 6: Architektur mit den von uns gewählten Frameworks 22](#_Toc132226808)

[Abbildung 7: ER-Diagramm der Datenbankstruktur 24](#_Toc132226809)

[Abbildung 8: Vorläufiges Klassendiagramm der Implementierung 24](#_Toc132226810)

[Abbildung 9: Ablaufdiagramm einer einzelnen Challenge 25](#_Toc132226811)

[Abbildung 10: Vorläufige Sprintplanung zum Stand 12. April 2023 26](#_Toc132226812)

1. [9] Öffentlich zugängliche Beschreibung der ISO-Norm 25010 [↑](#footnote-ref-2)
2. [8] Conventional Commits v1.0.0 Spezifikation [↑](#footnote-ref-3)