

# Interaktive Skisprung-Trainingsanalyse

## Bachelorarbeit

Abteilung Informatik  
Hochschule für Technik Rapperswil

Frühjahrssemester 2018

Autoren:	Janic Mikes, Marcel Maeder
Betreuer:	Prof. Dr. Markus Stolze
Projektpartner:	Swiss-Ski, Muri bei Bern
Experte:	Thomas Kälin
Gegenleser:	Prof. Dr. Andreas Steffen



# Abstract

## Ausgangslage

Swiss-Ski möchte ihre Skisprung-Trainings genauer auswerten. Sie haben dazu ein GPS-Messsystem in Auftrag gegeben, welches im Rahmen einer Bachelorarbeit der HSR Abteilung Elektrotechnik im Jahr 2017 umgesetzt wurde.

Die Daten, die mit der Lösung der Vorarbeit gesammelt werden, sollen nun für die Trainer und Athleten dargestellt werden können.

## Vorgehen und Technologien

Im ersten Schritt wurden die Anforderungen aufgenommen und Lösungen evaluiert. Danach wurden die Lösungsbestandteile in einem agilen Prozess implementiert.

Das Frontend wurde als Angular Webapplikation umgesetzt. Das Backend wurde mit Django realisiert. Dieses dient als API-Schnittstelle. Die API wurde gemäss dem OpenAPI Standard mit Swagger.io dokumentiert.

Das Design des Frontends folgt den Regeln von Material Design, welches häufig bei Android Geräten anzutreffen ist.

Die Darstellung einer Sprungtrajektorie wird dreidimensional mit Three.js und zweidimensional mit Chart.js umgesetzt.

## Ergebnis

Im Rahmen der Bachelorarbeit ist eine Plattform entstanden, welche für die Trainer und Athleten von Swiss-Ski auf ihrem Server produktiv verfügbar ist.

Es lassen sich Trainings, Athleten, Trainer und Sprünge verwalten und anzeigen. Das Dashboard ermöglicht es, den Sprung gleichzeitig in einem 2D Graph, einem 3D Graph und mit einer Videoaufnahme zu verfolgen. Ausserdem können mehrere Sprünge miteinander verglichen werden.

# Management Summary

- Ausgangslage** Swiss-Ski möchte ihre Skisprung-Trainings genauer auswerten. Sie haben dazu ein GPS-Messsystem in Auftrag gegeben, welches im Rahmen einer Bachelorarbeit der HSR Abteilung Elektrotechnik im Jahr 2017 umgesetzt wurde. Die Daten, die mit der Lösung der Vorarbeit gesammelt werden, sollen nun für die Trainer und Athleten dargestellt werden können. Eine Studentengruppe hat dafür bereits einen Prototypen geschrieben.
- Vorgehen und Technologien** Im ersten Schritt wurden die Anforderungen aufgenommen und Lösungen evaluiert. Danach wurden die Lösungsbestandteile in einem agilen Prozess implementiert.  
Die Lösung wurde als Webseite umgesetzt. Bei der Wahl der Technologien wurde insbesondere auf die Wartbarkeit geachtet.  
Das Design der Webseite folgt den Regeln von Material Design, welches häufig bei Android Geräten anzutreffen ist.
- Ergebnis** Es wurde eine Plattform entwickelt, welche für die Trainer und Athleten von Swiss-Ski nun verfügbar ist.  
Es lassen sich Trainings, Athleten, Trainer und Sprünge verwalten und anzeigen. Das Dashboard ermöglicht es, den Sprung gleichzeitig in einem 2D Graph, einem 3D Graph und mit einer Videoaufnahme zu verfolgen. Ausserdem können mehrere Sprünge miteinander verglichen werden.  
Bei einem Trainingsbesuch wurde die Applikation unter realen Bedingungen getestet (siehe Abbildung 0.1). Die Trainingssprünge konnten erfolgreich aufgezeichnet und eingespielt werden. Mit einem Trainer wurde ein User Acceptance Test durchgeführt.  
Die Webseite wurde für die Darstellung auf Notebooks optimiert, da es sich herauskristallisiert hat, dass ein Zugriff mit anderen Geräten für das Training nicht notwendig ist.
- Ausblick** Gewisse administrative Tätigkeiten, wie die Benutzerverwaltung, müssen über eine gesonderte Webseite getätigt werden. Diese Funktionalität könnte in die reguläre Webseite eingebaut werden. Eine automatische Erkennung vom Absprung und Landungspunkt würde die Sprungerfassung noch benutzerfreundlicher machen.



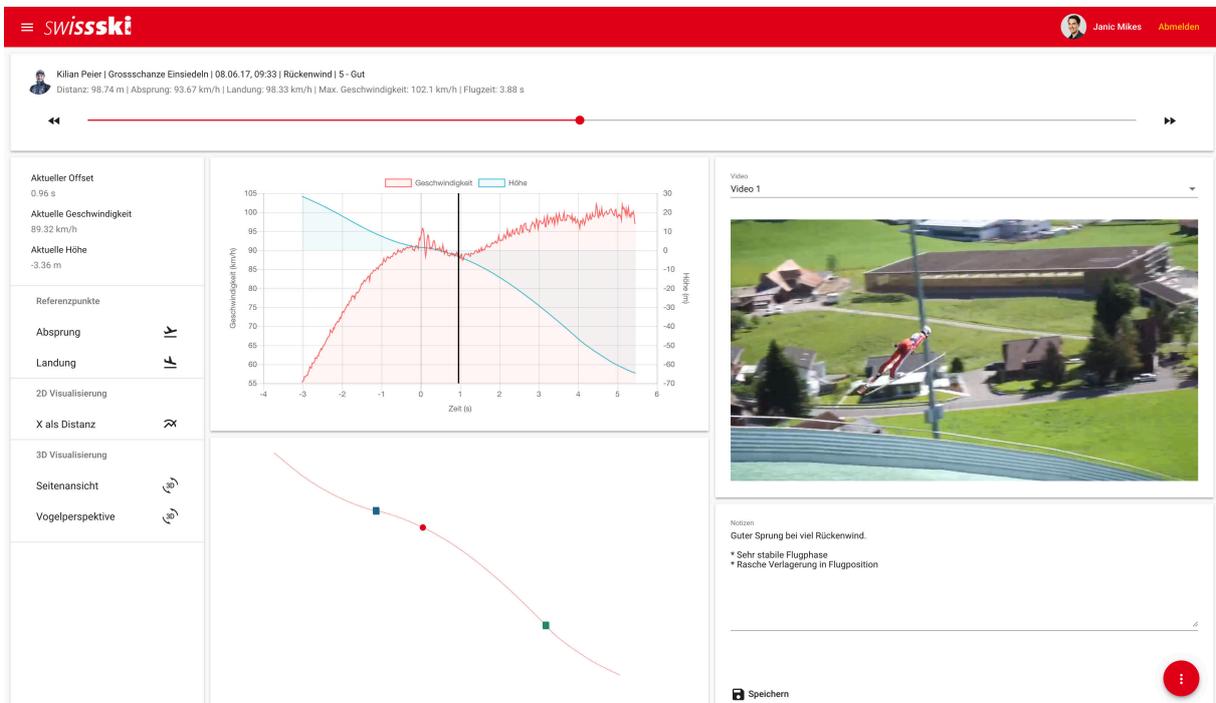
(a) Trainingsbesuch in Einsiedeln



(b) Einrichten der Messgeräte



(c) Panorama Sprungszance Einsiedeln



(d) Screenshot - Sprungdashboard

Abbildung 0.1: Eindrücke während der Bachelorarbeit

# Inhaltsverzeichnis

	Seite
<b>1 Einführung</b>	<b>9</b>
1.1 Problem . . . . .	9
1.2 Abgrenzung . . . . .	9
<b>2 Aufgabenstellung</b>	<b>10</b>
<b>3 Anforderungsanalyse</b>	<b>13</b>
3.1 Produkt Anforderungen . . . . .	13
3.1.1 Produkt Perspektive . . . . .	13
3.1.2 Produkt Funktion . . . . .	13
3.1.3 Benutzer Charakteristik . . . . .	13
3.1.4 Einschränkungen . . . . .	13
3.1.5 Annahmen . . . . .	14
3.1.6 Abhängigkeiten . . . . .	14
3.1.7 Randbedingungen . . . . .	14
3.2 Wichtige Use Cases . . . . .	14
3.2.1 Videos zu den Sprüngen hinzufügen . . . . .	14
3.2.2 Sprung Referenzpunkt festlegen . . . . .	15
3.2.3 Mehrere Sprünge vergleichen können . . . . .	17
3.3 Domainmodell . . . . .	17
3.4 UI Entwürfe . . . . .	17
3.5 Nicht funktionale Anforderungen . . . . .	17
3.5.1 Usability . . . . .	17
3.5.2 Accessibility . . . . .	17
3.5.3 Performance . . . . .	20
3.5.4 Availability . . . . .	20
3.5.5 Maintainability . . . . .	20
3.5.6 Testability . . . . .	20
3.5.7 Recovery . . . . .	21
3.5.8 Security and Privacy . . . . .	21
3.5.9 Portability . . . . .	21
3.5.10 Fault Tolerance . . . . .	22

<b>4</b>	<b>Architektur</b>	<b>23</b>
4.1	Designentscheide . . . . .	23
4.1.1	Frontend . . . . .	23
4.1.2	Backend . . . . .	26
4.1.3	Automatische Zeitsynchronisation . . . . .	27
4.2	Systemübersicht . . . . .	29
4.2.1	Environments . . . . .	29
4.2.2	Fronted . . . . .	29
4.2.3	API Server (Backend) . . . . .	29
4.3	Architektonische Ziele und Einschränkungen . . . . .	29
4.4	Logische Architektur . . . . .	31
4.4.1	Frontend . . . . .	31
4.4.2	Backend . . . . .	32
4.5	API Schnittstelle . . . . .	33
4.5.1	Wichtige Abläufe . . . . .	34
4.6	Prozesse und Threads . . . . .	34
4.7	Continuous Integration . . . . .	34
4.7.1	Linting . . . . .	34
4.7.2	Unit Testing . . . . .	34
4.8	Continuous Deployment . . . . .	34
4.8.1	Frontend . . . . .	35
4.8.2	Backend . . . . .	35
4.8.3	Deployment Diagramm . . . . .	35
4.9	Datenspeicherung . . . . .	35
4.10	Größen und Leistung . . . . .	35
4.11	Lizenzen . . . . .	35
4.11.1	Frontend . . . . .	36
4.11.2	Backend . . . . .	36
4.11.3	Erläuterung zu den Lizenzen . . . . .	37
<b>5</b>	<b>Implementation</b>	<b>40</b>
5.1	Prototypen . . . . .	40
5.2	Frontend . . . . .	41
5.2.1	Code Review . . . . .	41
5.3	Backend . . . . .	42
5.3.1	Code Review . . . . .	42
5.4	Code Metriken . . . . .	43
5.5	Security Review . . . . .	45
5.6	User Acceptance Test . . . . .	47
5.6.1	Vorbereitung . . . . .	47
5.6.2	Szenario . . . . .	47

<b>6</b>	<b>Ausblick</b>	<b>51</b>
6.1	Hohe Priorität . . . . .	51
6.2	Mittlere Priorität . . . . .	51
6.3	Geringe Priorität . . . . .	51
<b>7</b>	<b>Verzeichnisse</b>	<b>52</b>
7.1	Glossar und Abkürzungen . . . . .	52
7.2	Abbildungsverzeichnis . . . . .	55
7.3	Tabellenverzeichnis . . . . .	55
7.4	Literatur . . . . .	56
<b>8</b>	<b>Appendix</b>	<b>60</b>
8.1	Test Zielerreichung . . . . .	60
8.2	Use Cases . . . . .	65
8.3	UI Mockups . . . . .	72
8.4	Continuous Integration und Deployoment . . . . .	80
8.4.1	GitLab Runner . . . . .	80
8.4.2	Eigene Base Images . . . . .	80
8.4.3	Buildfiles . . . . .	82
8.5	Server Umgebung . . . . .	83
8.5.1	Dokku . . . . .	83
8.5.2	Aufsetzen Backend App . . . . .	83
8.5.3	Aufsetzen Client App . . . . .	85
8.5.4	Backup Restore . . . . .	86
8.5.5	Filewatcher . . . . .	86
8.6	Entwicklungsumgebung . . . . .	87
8.6.1	Frontend . . . . .	87
8.6.2	Backend . . . . .	88
8.7	Benutzeranleitung . . . . .	90
8.7.1	Trainingsverwaltung . . . . .	90
8.7.2	Sprungverwaltung . . . . .	91
8.8	Administratorenanleitung . . . . .	94
8.8.1	User verwalten . . . . .	94
8.8.2	Sprünge verwalten . . . . .	95
8.8.3	Schanzen verwalten . . . . .	95
8.8.4	Videos verwalten . . . . .	96
8.9	API Dokumentation . . . . .	97
8.10	NPM Audit . . . . .	128
8.11	Frontend Thirdparty Lizenzen . . . . .	133
8.12	PIL License . . . . .	142

# 1 Einführung

Swiss-Ski möchte Skisprung-Trainings genauer auswerten. Sie haben dazu ein Messsystem in Auftrag gegeben, welches im Rahmen einer Bachelorarbeit der HSR Abteilung Elektrotechnik umgesetzt wurde [1]. Die aufgezeichneten GPS Daten vom Messgerät aus der Vorarbeit sollen für die Trainer und Athleten dargestellt werden. Eine Studentengruppe von der HSR hat dafür einen Prototypen geschrieben. Dieser prüfte die Machbarkeit einer solchen Visualisierung mit HTML5 Webtechnologien.

## 1.1 Problem

Die Herausforderung dieser Arbeit besteht darin, die gegebenen Daten für die Benutzer in einer verwertbaren Form darzustellen. Ausserdem sind die bestehenden Anforderungen noch nicht genau spezifiziert. Es muss gemeinsam mit den Trainern eine für sie nützliche Lösung erarbeitet werden.

Weitere Herausforderungen sind die Darstellung von mehreren Sprüngen für einen Sprungvergleich und die gleichzeitige Darstellung des Sprunges mit der dazugehörigen Videoaufnahme.

## 1.2 Abgrenzung

Diese Arbeit befasst sich mit der Darstellung von aufgezeichneten GPS Messdaten. Das Messsystem, welches die Sprünge aufzeichnet, ist nicht Teil der Arbeit.

## 2 Aufgabenstellung

### Aufgabenstellung Bachelorarbeit "Interaktive Skisprung-Trainingsanalyse"

#### 1. Betreuer und Experte

Diese Arbeit wird für die Swiss Ski durchgeführt und wird von Prof. Dr. Markus Stolze, HSR, IFS, [mstolze@hsr.ch](mailto:mstolze@hsr.ch) betreut.

Industriepartner:

- Fabian Ammann, Swiss Ski

#### 2. Studierende

Diese Arbeit wird als Bachelorarbeit an der Abteilung Informatik durchgeführt von

- Janic Mikes
- Marcel Maeder

#### 3. Einführung

Das Projekt Skijumping hat zum Ziel, GPS Messdaten von Trainingssprüngen für die Trainer und Athleten aufzubereiten und die erhobenen Daten ins Training beziehungsweise in die Trainingsanalyse einfließen zu lassen. Dazu werden die Athleten während den Trainings mit GPS Messgeräten ausgestattet und die Daten an einen Server übermittelt.

#### 4. Ziele der Arbeit

Das Ziel dieser Arbeit ist es eine Web-Anwendung mit einem Client- und Serverteil zu realisieren, welche auf neue Messdaten (in Form von Positionsdateien) reagiert, sprich diese in eine Datenbank aufnimmt, per Backend zur Verfügung stellt und im Frontend visuell ansprechend präsentiert.

Die Positionsdateien enthalten einen Zeitstempel sowie die Raumkoordinaten im ECEF Format.

Es soll eine Web-Anwendung entstehen, die den Trainern und Athleten Informationen zu den Sprüngen bietet. Dazu sollen die Sprünge visualisiert werden, Informationen, wie Flughöhe, Geschwindigkeit etc., wenn möglich aus den Daten entnommen werden. Es soll möglich sein, Sprünge zu vergleichen und auch zusätzliche Angaben zu erfassen (z.B. Videos zu den Sprüngen hinzufügen).

Das Projekt Skijumping baut auf einer vorgängigen Bachelorarbeit auf, die sich mit Messen der Sprünge, einem ersten Vorprozessieren und dem zur Verfügung stellen der Daten auseinandergesetzt hat. Es kann davon ausgegangen werden, dass die Resultate aus der Vorarbeit in brauchbarer Form verfügbar sind. Änderungen an der vorherigen Arbeit sind nicht Teil dieser Arbeit. Sollten Änderungen nötig sein, ist dies mit den Verantwortlichen der vorgängigen Arbeit auszuhandeln.

Zusätzliche Funktionen können je nach Zeitbudget und Priorität des Auftraggebers umgesetzt werden.

Es finden wöchentliche Besprechungen mit dem Betreuer statt. Zusätzliche Besprechungen sind nach Bedarf durch die Studierenden zu veranlassen.

Alle Besprechungen, ausser der Kick-off Besprechung, sind von den Studierenden mit einer Traktandenliste vorzubereiten und zu leiten. Bei jedem Meeting soll der aktuelle Stand des Projektes präsentiert werden (Was wurde gemacht? Was wurde erreicht?). Die Beschlüsse der Besprechungen sind durch die Studierenden zu protokollieren und an den Betreuer anschliessend zuzustellen. Die Zeit-Protokollierung wird bei Bedarf oder nach Wunsch des Betreuers eingesehen.

Für die Durchführung der Arbeit ist ein Projektplan zu erstellen. Dabei ist auf einen kontinuierlichen und sichtbaren Arbeitsfortschritt zu achten. An Meilensteinen (oder auch Zwischenversionen) gemäss Projektplan sind einzelne Arbeitsergebnisse in vorläufigen Versionen abzugeben. Über die abgegebenen Arbeitsergebnisse erhalten die Studierenden ein vorläufiges Feedback. Eine definitive Beurteilung erfolgt auf Grund der am Abgabetermin abgelieferten Resultate.

## 5. Dokumentation

Über diese Arbeit ist eine Dokumentation gemäss den Richtlinien der Abteilung Informatik zu verfassen. Die zu erstellenden Dokumente bzw. Berichtsteile sind im Projektplan festzuhalten. Alle Dokumente sind nachzuführen, d.h. sie sollten den Stand der Arbeit bei der Abgabe in konsistenter Form dokumentieren.

## 6. Termine

Siehe auch Terminplan auf dem Skripteserver (Fachbereich/Bachelor-Arbeit\_Informatik/BAI/)

<b>Montag, den 19.02.2018</b>	<b>Beginn der Studienarbeit</b>
<b>08.06.2018</b>	<p>Die Studierenden geben den Abstract für die Diplomarbeitbroschüre zur Kontrolle an ihren Betreuer/Examinator frei. Die Studierenden erhalten vorgängig vom Studiengangsekretariat die Aufforderung mit den Zugangsdaten zur Online-Erfassung des Abstracts im DAB-Tool.</p> <p>Die Studierenden senden per Email das A0-Poster zur Prüfung an ihren Examinator/Betreuer.</p> <p>Vorlagen sowie eine ausführliche Anleitung betreffend Dokumentation stehen auf dem Skripteserver zur Verfügung.</p>
<b>13.06.2018</b>	<p>Der Betreuer/Examinator gibt das Dokument mit dem korrekten und vollständigen Abstract der Broschüre zur Weiterverarbeitung an das Studiengangsekretariat frei.</p> <p>Fertigstellung und Weitergabe des A0 Posters per Email bis 10.00 Uhr an das Studiengangsekretariat.</p>

<b>15.06.2018</b>	<b>Abgabe des Berichts an den Betreuer bis 12.00 Uhr</b>
	Präsentation und Ausstellung der Bachelorarbeiten, 16 bis 20 Uhr
<b>Bis zum 24.08.18</b>	Mündliche BA-Prüfung

## 7. Beurteilung

Eine erfolgreiche Bachelorarbeit zählt 12 ECTS-Punkte pro Studierenden. Für 1 ECTS Punkt ist eine Arbeitsleistung von 30 Stunden.

Für die Beurteilung ist der verantwortliche Dozent zuständig.

Gesichtspunkt	Gewicht
1. Organisation, Durchführung	1/6
2. Berichte (Abstract, Mgmt Summary, technischer u. persönliche Berichte) sowie Gliederung, Darstellung, Sprache der gesamten Dokumentation	1/6
3. Inhalt*)	3/6
4. Mündliche Prüfung	1/6

\*) Die Unterteilung und Gewichtung von 3. Inhalt wird im Laufe dieser Arbeit präzisiert (u.A. durch die Bewertungsmatrix)

Im Übrigen gelten die Bestimmungen der Abteilung Informatik für Studienarbeiten.

Rapperswil, den 12. April 2018.



Prof. Dr. Markus Stolze

## 3 Anforderungsanalyse

Im ersten Teil werden die allgemeinen Eigenschaften der Software erläutert. Danach folgt die Auflistung der Use Cases. Zum Schluss werden weitere Anforderungen besprochen.

### 3.1 Produkt Anforderungen

#### 3.1.1 Produkt Perspektive

Diese Software ist eine Neuentwicklung. Es gibt nach unserer Recherche keine andere Software, die diese Funktionalität bietet. Das Produkt soll nach dieser Arbeit von Swiss-Ski auf ihrem Server eingesetzt werden können.

Auf dem Server wurde dazu die Platform as a Service (PaaS) Lösung Dokku bereits installiert.

In Abbildung 3.1 befindet sich das Kontext Diagramm.

#### 3.1.2 Produkt Funktion

Das Produkt verfügt über folgende Hauptfunktionen:

- Visualisierung Sprungdaten
- Vergleich Sprungdaten
- Trainingsverwaltung
- Upload Video des Sprunges
- Reagieren auf neue Messdaten

#### 3.1.3 Benutzer Charakteristik

Zielgruppe des Produktes sind Skisprung-Trainer sowie deren Athleten. Es gibt fünf Trainer, die diese Applikation nutzen werden.

#### 3.1.4 Einschränkungen

Das Produkt ist auf akkurate Daten des GPS-Messsystems angewiesen.

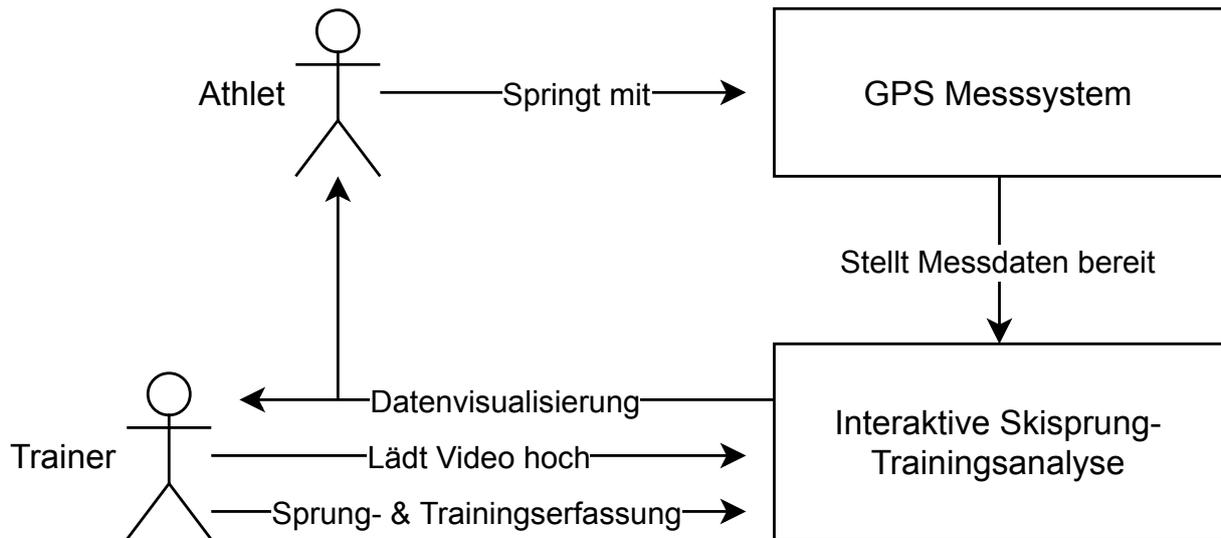


Abbildung 3.1: Kontext Diagramm

### 3.1.5 Annahmen

Es ist eine schnelle Internetverbindung an den Verwendungsorten vorhanden. Sprungdaten werden in einem definierten Ordner bereitgestellt.

### 3.1.6 Abhängigkeiten

Verfügbarkeit der Messdaten mehrerer Sprünge während der Entwicklung. Ohne Messdaten kann das System nur eingeschränkt getestet werden.

### 3.1.7 Randbedingungen

Das Produkt muss auf dem Server [gps.skispringen.ch](https://gps.skispringen.ch) (Cloudscale Virtual Private Server (VPS)) mit dem Dokku-System deployed werden.

## 3.2 Wichtige Use Cases

Nachfolgend sind drei wichtige Use Cases aufgeführt, die eine zentrale Rolle in unserer Applikation haben. Die Abbildung 3.2 und 3.3 zeigen die Use Cases als Diagramm. Die gesamte Liste der Use Cases befindet sich im Anhang 8.2.

### 3.2.1 Videos zu den Sprüngen hinzufügen

#### Voraussetzungen

- Der Sprung muss erfasst sein. Das Video ist noch keinem Sprung zugeordnet

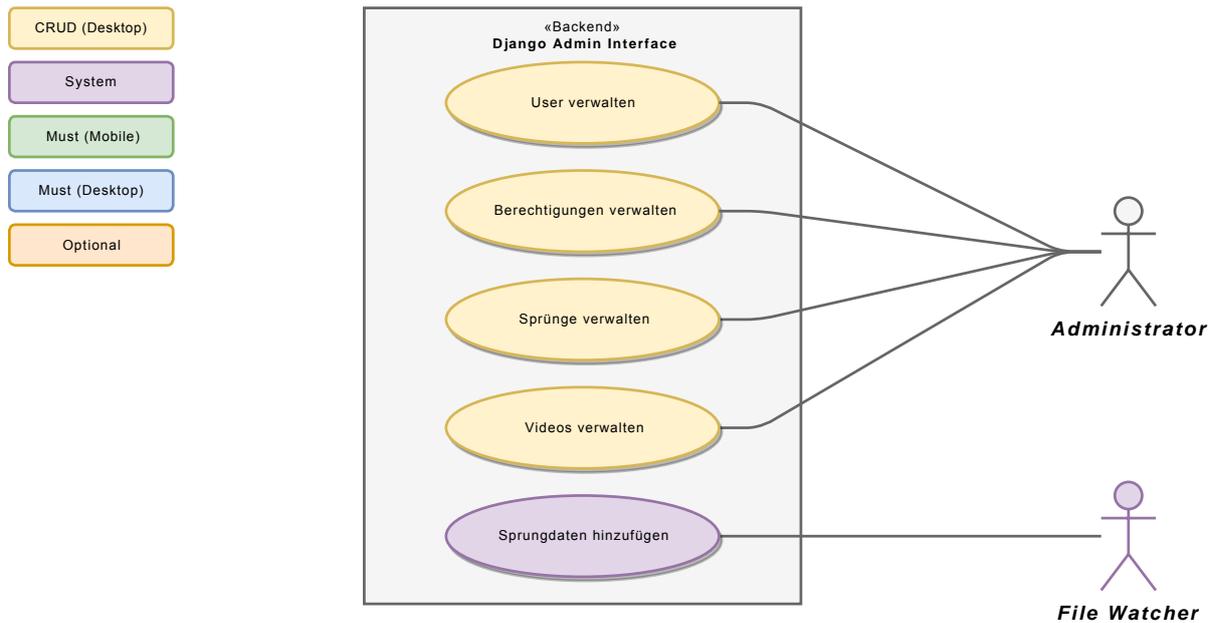


Abbildung 3.2: Use Case Diagramm (Administrativ)

### Erfolgreiches Szenario

Beim Ansehen eines Sprunges kann der Trainer aus einer Liste ein Video auswählen und dieses dem Sprung zuweisen. Post Conditions Das Video ist dem Sprung zugeordnet.

### Erweiterungen

Anstelle der Auswahlliste kann ein Video direkt hochgeladen werden.

Optional: Beim Hinzufügen von einem Video können auch bereits zugewiesene Videos angezeigt werden. Diese werden dann vom alten Sprung entfernt und neu zugewiesen.

## 3.2.2 Sprung Referenzpunkt festlegen

### Aktoren

Trainer

### Voraussetzungen

- Der Sprung muss erfasst sein.

### Erfolgreiches Szenario

Beim Anzeigen des Sprunges kann dessen Referenzmesspunkt (Schanzentisch, Absprung) festgelegt werden. Dieser Punkt wird mit dem Sprung gespeichert.

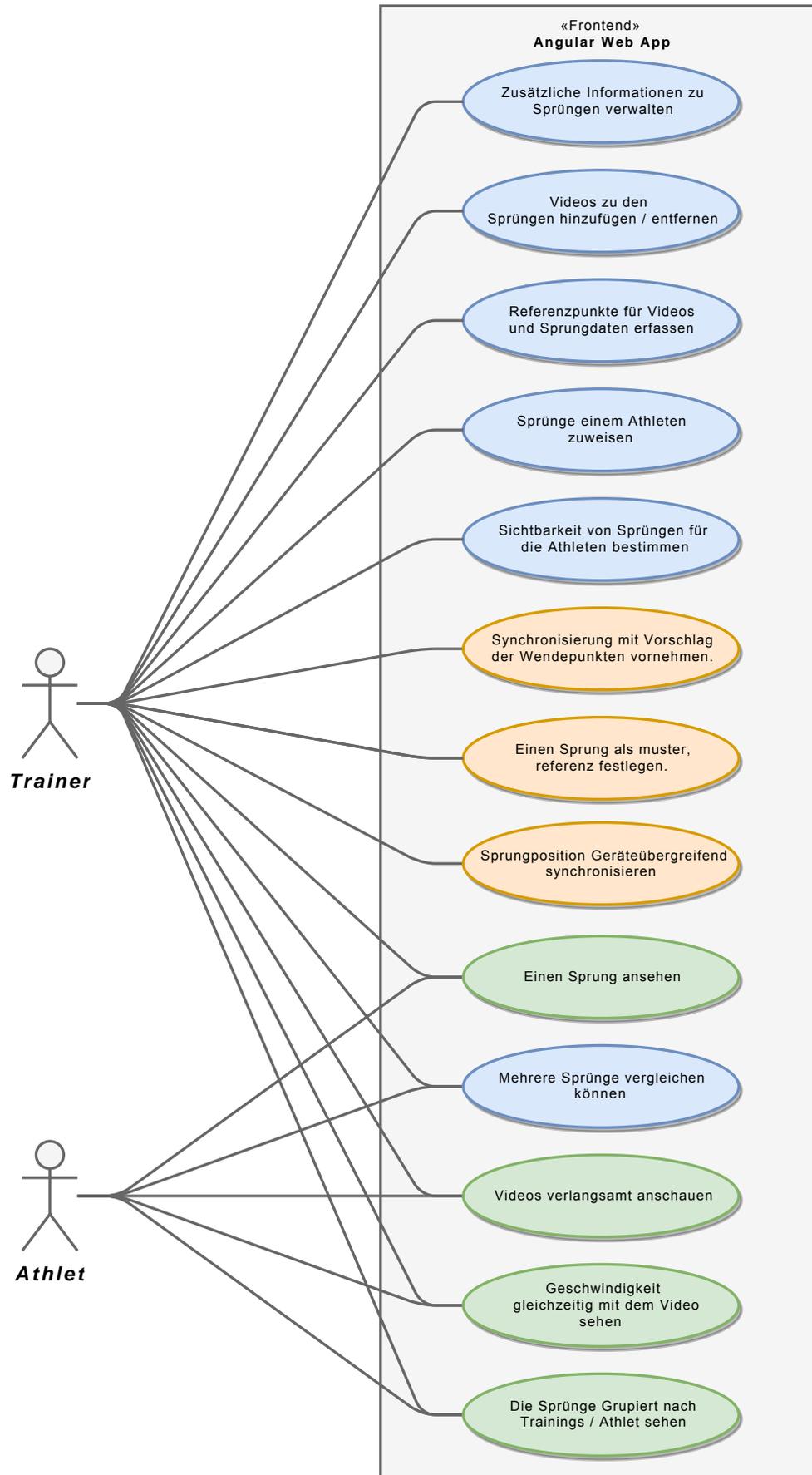


Abbildung 3.3: Use Case Diagramm

### 3.2.3 Mehrere Sprünge vergleichen können

#### Aktoren

Trainer, Athlet

#### Voraussetzungen

- Alle Sprünge, welche verglichen werden sollen, müssen erfasst sein.

#### Erfolgreiches Szenario

Beim Ansehen eines Sprunges können weitere Sprünge als Vergleich hinzugefügt werden. Die Flugbahnen werden in verschiedenen Farben gleichzeitig dargestellt.

## 3.3 Domainmodell

Das Domainmodell ist in Abbildung 3.4 zu sehen.

## 3.4 UI Entwürfe

Die Abbildung 3.5 zeigt die vier wichtigsten UI Mockups. Sie sind die Grundlage für unser Frontend UI. Im Anhang befindet sich der vollständige UI Mockupkatalog im Abschnitt 8.3. Die Mockups wurden mit Balsamiq erstellt [2].

## 3.5 Nicht funktionale Anforderungen

### 3.5.1 Usability

Die Applikation wird von einer kleinen Gruppe von Anwendern verwendet.

Es wird ein Usability Test mit Testpersonen durchgeführt. Das Ergebnis wird mit einem Testprotokoll dokumentiert.

Die Material Designrichtlinien werden eingehalten.

### 3.5.2 Accessibility

Es ist jedoch bei der Farbauswahl darauf zu achten, dass Menschen mit Rot-grün-Sehschwäche die Applikation trotzdem benutzen können.

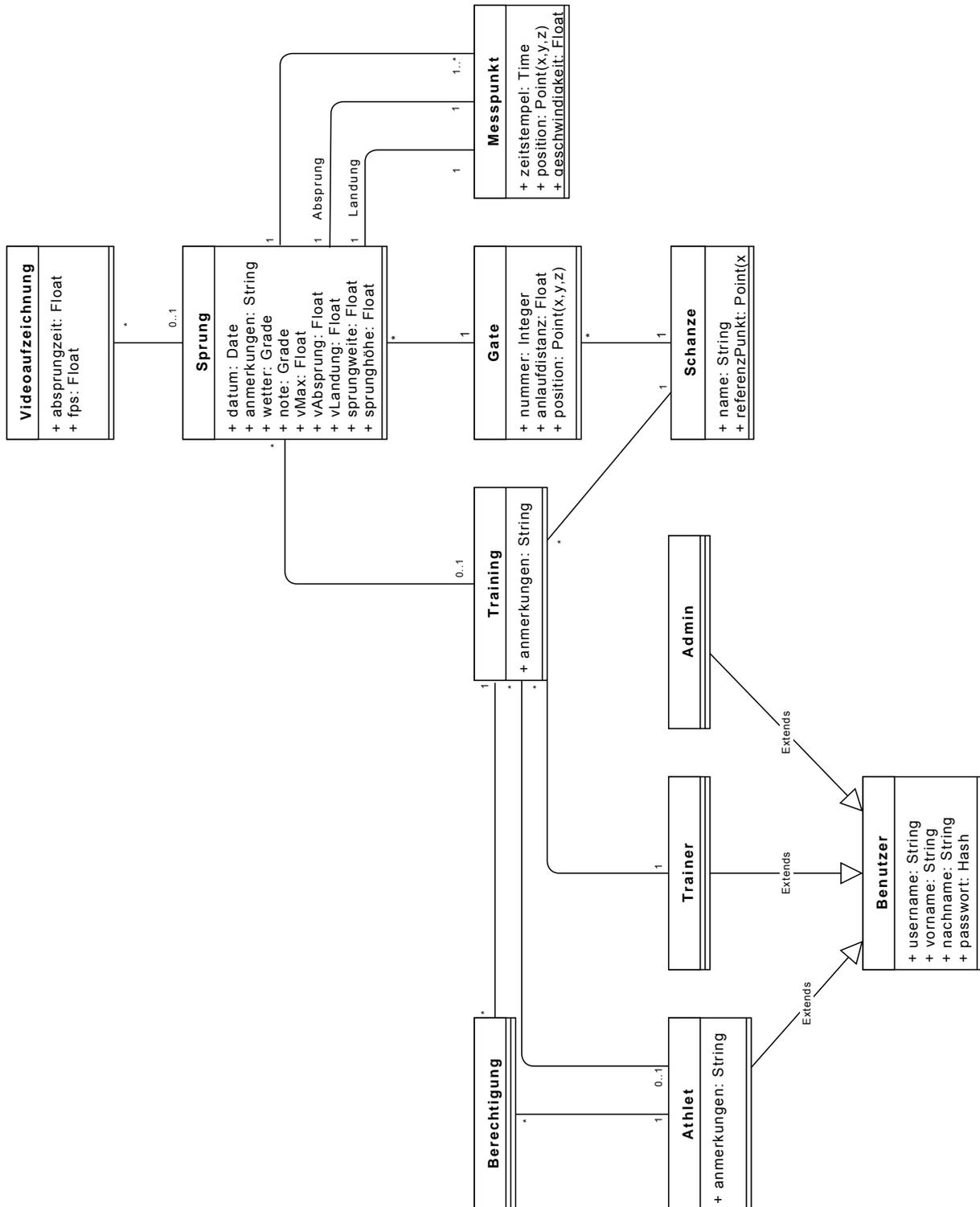
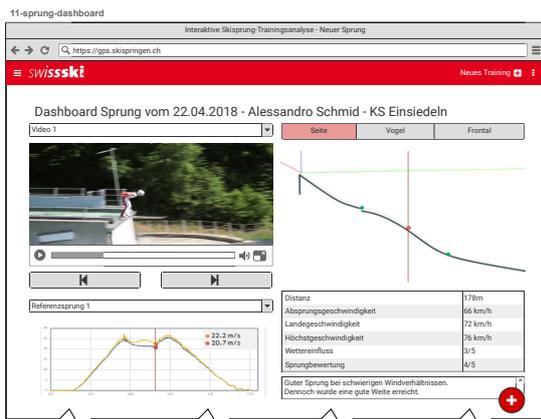
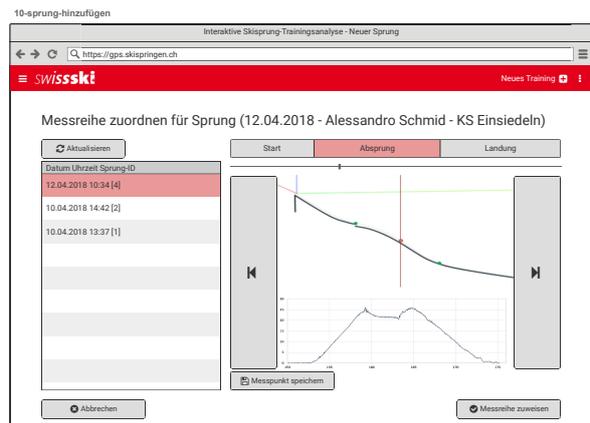


Abbildung 3.4: Domainmodell



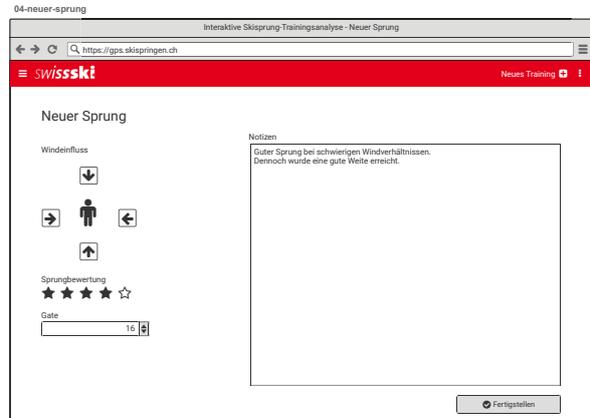
(a) Dashboard UI Entwurf



(b) Messreihe zuordnen UI Entwurf



(c) Neues Video hochladen UI Entwurf



(d) Neuer Sprung

Abbildung 3.5: UI Entwürfe aus Balsamiq für Confluence

### 3.5.3 Performance

Es gibt keine Anforderung auf die Dauer der Datenübertragung.

Nachdem im Inputordner ein File des Messsystems zur Verfügung steht, darf es maximal 2 Minuten gehen, bis der Sprung über die API abgerufen werden kann.

### 3.5.4 Availability

Es gibt keine Anforderungen an die Verfügbarkeit. Die Applikation wird auf einem cloudscale.ch VPS betrieben.

### 3.5.5 Maintainability

#### Frontend

Die Coding Guidelines von angular.io [3] werden eingehalten und mit Reviews der Applikation geprüft. Das Ergebnis wird in schriftlicher Form festgehalten. Abweichungen sind zu Begründen. Der Sourcecode wird mit TS Lint [4] überprüft und hält die Regeln gemäss Angular CLI Standardkonfiguration ein.

#### Backend

Die Coding Guidelines von Django werden eingehalten und mit Reviews der Applikation geprüft. Das Ergebnis wird in schriftlicher Form festgehalten. Abweichungen sind zu Begründen.

Der Sourcecode wird mit pycodestyle (ehem. Pep8) [5] überprüft und hält den Regeln in angepasster Form stand. Die maximale Zeilenlänge wurde aus Lesbarkeitsgründen von 80 auf 120 erhöht.

#### Reviews

Reviews werden für Merge Requests auf Verlangen durchgeführt und protokolliert.

#### Anleitungen

Es gibt ein Benutzerhandbuch für die Endbenutzer, für die Administratoren wie auch für die Entwickler (siehe Anhang 8.7 und 8.6).

### 3.5.6 Testability

#### Frontend

Die wichtigsten Komponenten werden im Verlauf des Projekts spezifiziert und mit Unit Tests geprüft.

Im Frontend werden drei End to End (Integration Tests) mit Protractor implementiert. Sie simulieren eine Benutzerinteraktion mit der Applikation.

## Backend

Die wichtigsten Komponenten werden im Verlauf des Projektes spezifiziert und mit Unit Tests geprüft.

### 3.5.7 Recovery

Backups können über die integrierten Postgres Datenbankschnittstelle durchgeführt werden.

Es gibt eine Anleitung zur Wiederherstellung der Dokku-Applikationen (siehe Abschnitt 8.5.4).

### 3.5.8 Security and Privacy

Es gibt ein Review, welches die Applikation gegen die aktuellen OWASP Top Ten Schwachstellen überprüft. Dieses wird mit einem Reviewprotokoll festgehalten (siehe Abschnitt 5.5).

#### Authentication

Username und Passwort Authentisierung. Kennwörter werden von Django mit PBKDF2 gespeichert [6].

#### Authorisation

Admin und Trainer können Berechtigungen an die Athleten geben oder entziehen.

#### Privacy

Ein Sprung kann nur angesehen werden, wenn der Benutzer die Berechtigung dazu hat. Administratoren können alle Sprünge ansehen, Athleten nur ihre eigenen und Referenzsprünge.

#### Encryption

Transportverschlüsselung über SSL. Keine zusätzliche Verschlüsselung der Nutzdaten. Passwörter werden mit PBKDF2 gespeichert.

### 3.5.9 Portability

Die Software unterstützt die Version 67 von Google Chrome auf einem Laptop oder der aktuellen iOS Version mit dem Safari Browser auf einem Apple iPad Pro 10.5 Zoll.

Die Applikation ist nicht für Smartphones optimiert.

### **3.5.10 Fault Tolerance**

Bei auftretenden Fehler werden dem Benutzer diese dargestellt.

## 4 Architektur

Dieser Teil der Arbeit beschreibt die erarbeitete Softwarearchitektur.

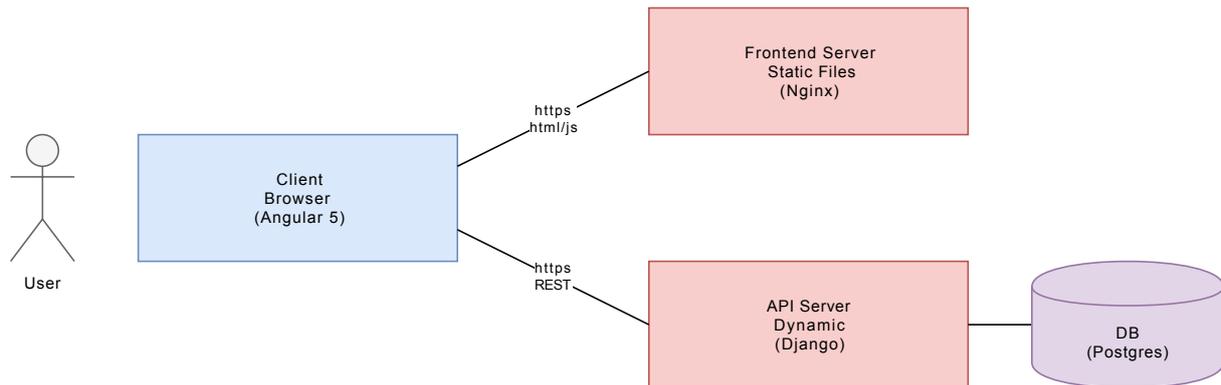


Abbildung 4.1: Systemübersicht

### 4.1 Designentscheide

#### 4.1.1 Frontend

##### Frontend Framework

Vorgegeben ist die Verwendung von Angular oder React als Frontend Framework. Es soll typsicheres Javascript verwendet werden, um die Availability zu erhöhen. Hierfür wird TypeScript als bestgeeignete Lösung betrachtet. Flow geht zwar in die gleiche Richtung, hat aber noch nicht dieselbe Maturität.

Für die UI Elemente soll Material Design verwendet werden.

##### Angular

Webseite [7], Material Design [8], TypeScript [9] [10]

##### Positiv

- CLI Tools vorhanden
- Offizielle Material Design Unterstützung
- Bekannt aus HSR Modul WED3
- TypeScript standardmässig verwendet
- Trennung von Logik, HTML und SCSS files

Negativ

- Struktur stark vorgegeben
- Sehr viele Dependencies (*node\_modules* Ordner ist etwa 650 MByte gross und beinhaltet über 100'000 Dateien)

## React

Webseite [11], Material Design [12], TypeScript [13] [14]

Positiv

- CLI Tools vorhanden
- Bekannt aus HSR Modul WED3

Negativ

- Material Design nur mit Third-Party Package möglich
- TypeScript muss selber aufgesetzt / konfiguriert werden mit Webpack
- Vermischung von Code und View in JSX/TSX Dateien

## Entscheid

Die Entscheidung wurde zugunsten von Angular gefällt.

Die Hauptgründe sind das offizielle Material Design Projekt, die standardmässige Verwendung von TypeScript sowie die strikte Trennung von HTML und SCSS Dateien im Code.

Vom funktionsumfang her könnte das Projekt mit beiden Frameworks erfolgreich umgesetzt werden.

## Visualisierungsframework

Die Anforderungen ergeben sich aus dem Use Case «Einen Sprung ansehen» im Abschnitt 8.2.

3D: Die Ansicht, Perspektive soll gespeichert und wieder abgerufen werden können (Sicht von Oben und von der Seite).

2D: Das Framework soll ein Zoom & Pan unterstützen, um mehr Details sehen zu können.

3D & 2D: Ein einzelner Punkt "X" muss als aktiv gekennzeichnet werden können, mit einer Linie oder einem Ball (Highlighting).

### 3D - three.js

Webseite [15], Kamera Perspektive [16], Darstellen einer Linie [17], Hervorheben eines Punktes [18]

- Viele Beispiele ohne ausführliche Dokumentation
- Scene und Kamera Perspektive müssen selber konfiguriert werden. Können persistiert werden.
- Bestehender Prototyp zeigt eine gute Eignung für unser Projekt. Kann mit vielen Datenpunkten umgehen.

### 3D - Vis.js

Webseite [19], Kamera Perspektive [20] [21], Darstellen: [22]

Hervorheben: Einzige Möglichkeit wäre ein Tooltip, diese lassen sich aber nicht imperativ steuern (Simulierte Maus-Interaktionen wurden nicht ausprobiert). Es können auch nicht zur Laufzeit neue Objekte über die Library auf den Canvas gezeichnet werden, womit ein Ball auch nicht in Frage kommt.

Aufgrund von diesen Mängeln ist Vis.js nicht für dieses Projekt geeignet.

- wurde bereits im bestehenden Prototyp verwendet
- Darstellung wird automatisch gemacht
- Imperatives hervorheben eines Punktes nicht möglich.
- Perspektive Kamera kann gespeichert und wiederhergestellt werden.

### 2D - Chart.js

Webseite [23], Darstellung [24], Zoom & Pan [25], Hervorheben [26]

- Plugin für Zoom vorhanden
- Plugin für die Darstellung von zusätzlichen Informationen vorhanden

### 2D - Vis.js

Darstellung [27]

- Wurde bereits im UI Prototyp verwendet
- Imperatives hervorheben eines Punktes nicht möglich.

### Entscheid

Für die 3D Visualisierung fällt der Entscheid auf `three.js` aufgrund der starken Anpassungsfähigkeit. Ausserdem wird `three.js` als geeignet für die Darstellung von Geodaten angesehen [28]

Für die 2D Darstellung wird `Chart.js` mit den Plugins für Zoom & Pan und Annotationen eingesetzt.

### Verwendete Libraries

Die im Frontend verwendeten Libraries sind in Tabelle 4.1 aufgeführt.

Library	Version	Begründung
Angular	5.2.10	Siehe Vergleich 4.1.1
Angular Material	5.2.3	Offizielles Package von Angular
Angular CDK	5.2.3	Verwendet von Angular Material
Angular Flex-Layout	5.0.0-beta.13	Ermöglicht die Verwendung von Flex-Layouts mit directives
Chart.js	2.7.2	Siehe Vergleich 4.1.1
chartjs-plugin-annotation	0.5.7	Ermöglicht das Darstellen von zusätzlichen Informationen
chartjs-plugin-zoom	0.6.3	Ermöglicht Zoom & Pan im Diagramm
three.js	0.91.0	Siehe Vergleich 4.1.1
zone.js	0.8.26	Standard Angular CLI
hammerjs	2.0.8	Verwendet von Angular Material
core-js	2.4.1	Standard Angular CLI
Material-Design-Icons	3.0.1	Verwendet von Angular Material
Roboto Fontface	0.9.0	Empfohlene Schrift für Material Design
RxJS	5.5.10	Standard Angular CLI
web-animations-js	2.3.1	Verwendet von Angular Material

Tabelle 4.1: Libraries im Frontend

## 4.1.2 Backend

### Backend Framework

Die Aufgabenstellung lässt die Wahl zwischen Django und NodeJS offen.

#### NodeJS

Webseite [29], Express-Framework [30]

##### Positiv

- Bekannt aus HSR Modul WED2
- Gleiche Programmiersprache wie Frontend

##### Negativ

- Kein OR Mapper integriert

#### Django

Webseite [31], Django Rest Framework (DRF) [32]

##### Positiv

- Bestehender Prototyp bereits mit DRF umgesetzt
- OR Mapper bereits integriert
- CoreAPI wird von DRF unterstützt

- Admin Interface einfach zu implementieren

Negativ

- Package Manager Pipenv nicht so ausgereift wie Node Package Manager (NPM)

### Entscheid

Es wird Django eingesetzt, weil der bestehende Prototyp bereits damit umgesetzt wurde. Technisch gesehen sind beides valable Lösungen.

### Backend Datenbank

Bezüglich Datenbank gibt es keine Vorgaben aus der Aufgabenstellung. Die Daten sind strukturiert. Deshalb werden nur relationale Datenbanken verglichen, welche von Django für produktive Umgebungen unterstützt werden und unter einer Open Source Lizenz verfügbar sind.

#### MariaDB

Webseite [33]

- Limitierte Funktionalität mit Django Spatial Database [34]
- Dokku Plugin vorhanden

#### PostgreSQL

Webseite [35]

- Prototyp verwendet PostgreSQL
- Bekannt aus HSR Modul DB1
- PostGIS GeoDB Integration
- Dokku Plugin vorhanden

### Entscheid

Eingesetzt wird PostgreSQL mit PostGIS Extension, weil der Prototyp bereits darauf basierte und PostgreSQL Teil von unserem Studium war.

### Verwendete Libraries

In der Tabelle 4.2 sind die verwendeten Backend Python Libraries aufgeführt. Die letzten drei Libraries wurden zur Erstellung der API Dokumentation im Anhang (siehe 8.9) verwendet. Sie sind für den Betrieb des Systems nicht zwingend notwendig.

#### 4.1.3 Automatische Zeitsynchronisation

Das System soll möglichst automatisch den Absprungpunkt in den Messdaten und im Video erkennen.

Wenn im Videofile ein genauer Zeitstempel des Aufnahmezeitpunktes wäre, könnte man den Offset zum Absprungpunkt direkt berechnen.

Library	Version	Begründung
Django Rest Framework	3.7.7	Siehe Vergleich 4.1.2
Django	2.0.3	Django Rest Framework
Coreapi	2.3.3	API Explorer unterstützt Entwickler
Gunicorn	19.7.1	Für <u>Dokku</u> benötigt
Django-Heroku	0.3.1	Für <u>Dokku</u> benötigt
Django-Cors-Headers	2.2.0	Benötigt da Angular App auf anderer Domain gehostet wird
Pillow	5.0.0	Django benötigt Pillow für ImageFields
drf-yasg	1.7.4	Swagger zur dynamischen Dokumentation der API
flex	6.13.1	Empfohlen für Swagger Plugin
swagger-spec-validator	2.1.0	Empfohlen für Swagger Plugin

Tabelle 4.2: Libraries im Backend

## Produkte mit GPS Unterstützung

### Sony

Keine Kamera im Consumer Bereich gefunden. Letzte Erwähnung von GPS Funktionalität in Marketingmaterial aus dem Jahr 2009. Im Professional Bereich gibt es das Modell PXW-X200 (CHF 6000), diese hat GPS integriert und zeichnet auch durchgängig Koordinaten im Stream auf. Zeitsynchronisation ist im Manual jedoch nicht erwähnt.

### Canon

Vom Hersteller Canon gibt es als Zubehörprodukt den GPS Empfänger GP-E2. Zu diesem sind folgende Kameras kompatibel:

- Legria HF G40. Total 1400.- Es sind jedoch nicht alle Features des GPS verfügbar. Zeitsynchronisation nur beim Einstecken des GPS Empfängers. [36]
- HF G26, nicht alle Features verfügbar [37]
- GX10, nicht alle Features verfügbar [38]

Alle unterstützen jedoch die Synchronisation der Uhrzeit des Camcorders bei Anschluss des GPS Empfängers. Auf dem Video selbst wird die Geoposition bei Aufnahmebeginn gespeichert.

### JVC

Keine Kameras mit GPS Funktion im Consumer Bereich gefunden. Sportkamera für \$7000 Listenpreis, diese ist speziell für Aufnahmen von American-Football Trainings gedacht.

## Andere Arten von Zeitstempel

Webseite mit der aktuellen Zeit als QR Code. Der Trainer nimmt nach dem Start der Aufnahme ein paar Sekunden diese Seite auf und wechselt danach zur Aufnahme des Sprunges.

Mögliche Probleme

- Ist die QR Seite am Aufnahmepunkt ersichtlich?
- Wie oft kann der QR Code wechseln, damit er trotzdem noch erkannt werden kann?
- Reicht die Genauigkeit?

## Fazit

Die Automatisierung der Videoabsprungpunkterkennung wird aus Zeitgründen nicht weiter verfolgt. Die Synchronisation erfolgt manuell durch den Benutzer.

## 4.2 Systemübersicht

Die Systemübersicht ist in Abbildung 4.1 dargestellt. Das Handling von Static-Files wird in Dokku automatisch mit Nginx gelöst.

### 4.2.1 Environments

### 4.2.2 Fronted

Für das Frontend wird eine Angular Single Page Application (SPA) eingesetzt.

### 4.2.3 API Server (Backend)

Als API Server wird Django zusammen mit dem Django Rest Framework verwendet.

## Datenbank

In der PostgreSQL Datenbank werden die Sprunginformationen wie auch Stammdaten der Athleten persistiert.

## Client

Clients sind Webbrowser, die der Funktionalität von Google Chrome auf PC Betriebssystemen entsprechen. Die zu unterstützende Version ist in den Nicht-Funktionalen-Anforderungen enthalten.

## 4.3 Architektonische Ziele und Einschränkungen

Für die Kommunikation zwischen Client und Server wird eine REST Schnittstelle eingesetzt, da diese eine Adaption auf weitere Plattformen in Zukunft zulässt.

Die Übertragung zwischen Client und Server wird mittels TLS verschlüsselt. Die Authentizität des Servers wird mittels eines Zertifikates von Let's encrypt sichergestellt. Eine zusätzliche Verschlüsselung der Daten in der Datenbank ist aus Sicht des Datenschutzes nicht

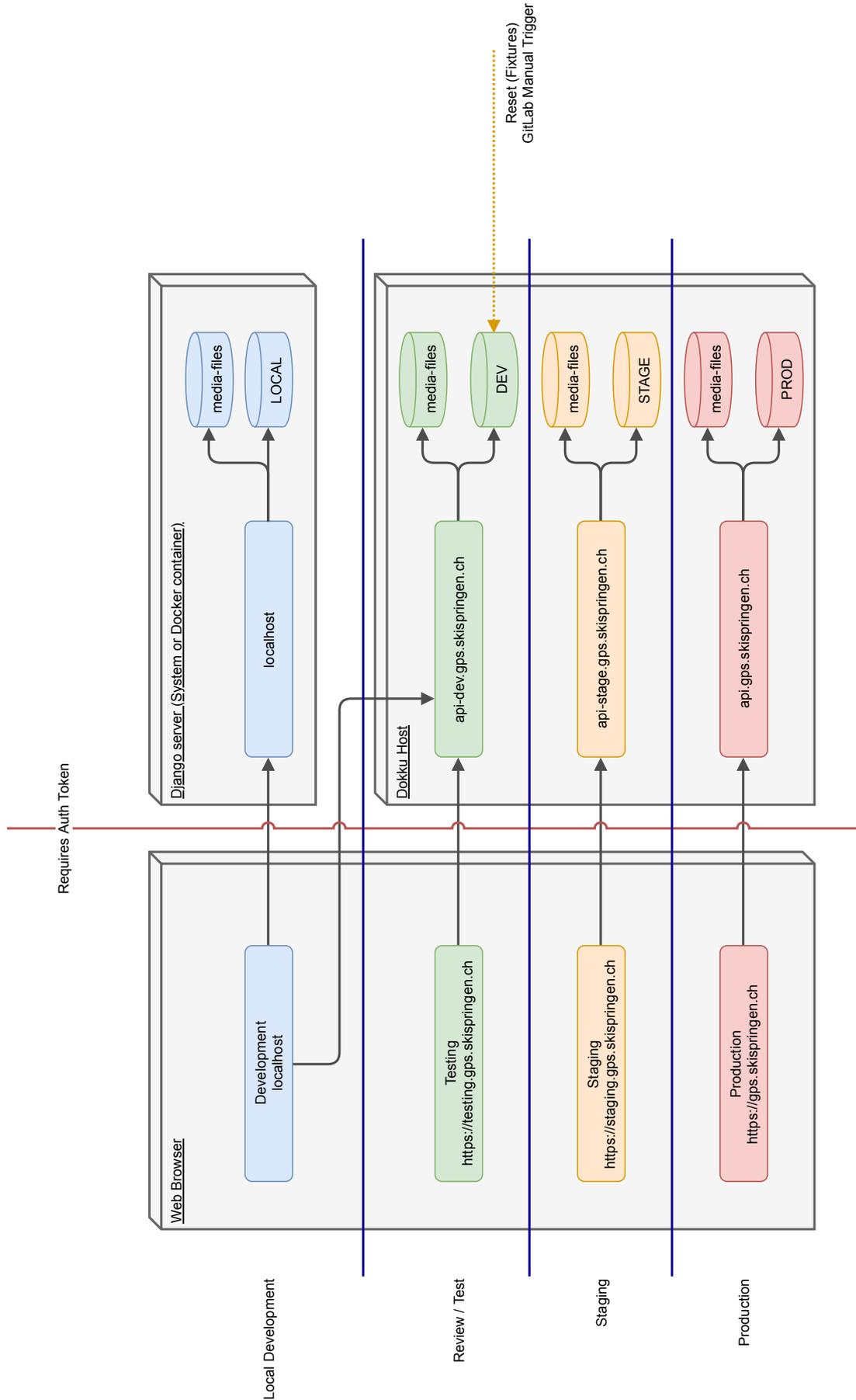


Abbildung 4.2: Environments

notwendig, da jeder Zugriff auf die Daten bereits authentisiert sein muss. Zudem speichert unser System keine besonders schützenswerte Daten.

## 4.4 Logische Architektur

### 4.4.1 Frontend

#### Ordnerstruktur

Die Ordnerstruktur entspricht den Guidelines von Angular [9]. Wichtig ist, dass man die bestandteile im Code einfach wiederfinden kann.

Insgesamt wurde die Applikation in folgende sechs Module unterteilt, welche die Komponenten, Services und mehr in deren Gruppierung unterteilen.

#### Auth

Im Auth-Module wird alles rund um die Authentifizierung gehalten. Auch der HTTP-Interceptor, welcher den Token an die Requests anhängt, befindet sich in diesem Modul. Zudem beinhaltet es Guards, welche die Zugriffe innerhalb der App kontrollieren. Die abstrakte Klasse mit den http Methoden befindet sich ebenfalls hier, diese wird von den spezifischen Ressourcen um deren Endpoints erweitert.

#### Core

Im Core-Module befinden sich appspezifische Kernkomponenten wie die Navigation, der Spinner, welcher eine Ladeaktivität anzeigt, oder auch die Not-Found-Komponente.

#### Jump

Das Jump-Module beinhaltet all jene Komponenten, welche direkt mit einem Sprung zu tun haben. Hier sind auch die Directives, welche einen Sprung visualisieren können.

#### Shared

Im Shared-Module sind jene Komponenten, welche auch von einer anderen App verwendet werden können. Hier befinden sich die Models, welche die Abstrahierung der Daten kennen.

#### Test

Das Test-Module wird nur für die Tests verwendet. Es übersteuert sämtliche in der App verwendete Ressourcen mit den Mock-Ressourcen, um die Services testbar zu machen. In diesem Ordner befindet sich auch die MockData Klasse, welche die Mockdaten enthält und bereitstellt.

#### Training

Im Training-Module ist die Trainingsverwaltung und Übersichtsseite gruppiert. Diese könnte so auch bei anderen Projekten verwendet werden.

#### Video

Das Video-Module beinhaltet eine Directive, welche Videos synchron zu einem externen Offset abspielen kann. Ausserdem beinhaltet sie die Komponenten für das Hochladen von Videodateien. Die Dashboardkomponente wird vom Jump-Modul wieder importiert, um ein Video auf dem Dashboard anzuzeigen. Wegen dieser Bindung hätte man das Video Modul auch in das Jump-Module integrieren können. Es wird die Exportfunktionen von Angu-

lar genutzt, welches diese Komponente in anderen Modulen zur Verfügung stellt, um die Orientierung im Code zu vereinfachen.

## Eingesetzte Patterns

### Observer Pattern

Alle Services werden mit Subscriptions gemacht. Anfragen an die API werden asynchron gemacht und beim Erhalt der Antwort weiterverarbeitet.

### MV(C)

Models und Views sind zwar vorhanden, der Controller ist aber nicht mehr ein Controller im eigentlichen Sinne. Es hat Ähnlichkeiten mit dem MVVM Pattern aus .NET, da mit Data-Binding gearbeitet wird.

## 4.4.2 Backend

### Eingesetzte Patterns

#### Facade Pattern

Die gesamte zur Verfügung gestellte REST-Schnittstelle entspricht dem Facade Pattern, da sich der Aufrufer nicht um die gesamte Komplexität kümmern muss.

#### Datenmodell

Die Abhängigkeiten der Django Model-Klassen werden in Abbildung 4.3 aufgezeigt. Das Diagramm wurde mit PyCharm erstellt.

#### Ordnerstruktur

Die Struktur basiert auf dem Django *startproject* Template.

```
1 / # Backend Repository
2 /fixtures/ # Datenbank Initialisierung
3 /scripts/ # Helperscripts (Filewatcher, Posfile Uploader)
4 /skiapi # SkiApi App
5   /migrations # DB Migrationen
6   /tests
7     /helpers # Object Factory, repetitive Tasks
8     conftest.py # py.test Globals
9     test_... # Tests
10  admin.py # Admin Interface
11  apps.py # App Konfiguration
12  models.py # Datenmodell / OR Mapper Konfiguration
13  permissions.py # Berechtigungs Prädikate
14  serializers.py # REST Felddefinitionen
15  services.py # Business Logik
16  signals.py # Django Signal Handler
17  swagger.py # Swagger API Dokumentation
18  urls.py # Routerkonfiguration
```

```
19     views.py           # URL Endpunkte
20     /skijumpapi/      # SkiJumpApi Projekt
21     settings.py      # Projektkonfiguration
22     urls.py          # Routerkonfiguration
23     wsgi.py          # Webserver Schnittstelle
24     /static/         # Static file (leerer Ordner)
25     /templates/      # Überschriebene Templates
26     .gitignore
27     .gitlab-ci.yml   # Continuous Integration
28     app.json         # Heroku DB Migrationen aktivieren
29     Dockerfile-dev   # Python 3.6.4 Dockerfile inkl. GEO Libraries
30     manage.py        # Django Project Manager
31     Pipfile          # Projekt Abhängigkeiten
32     Pipfile.lock     # Aktueller Stand der Abhängigkeiten
33     Procfile         # Heroku Start Webserver
34     pytest.ini       # Unit Test Konfiguration
35     README.md
36     setup.cfg        # Linting und Code Coverage konfiguration
```

Listing 4.1: Backend Ordnerstruktur

## Schnittstellen

- RESTful HTTP Schnittstelle
- API Explorer (Dev Facing)
- Admin Interface (User Facing)

## Wichtige interne Abläufe

### GraphPoints erstellen

Im Backend wird zwischen *MeasuringPoints* und *GraphPoints* unterschieden. *MeasuringPoints* gehören einem *Trajectory*. *GraphPoints* einem *Jump*. Wird einem *Jump* ein *Trajectory* zugewiesen, rechnet das Backend alle *MeasuringPoints* in *GraphPoints* um. Von den Koordinaten wird die Position des Schanzentisches abgezogen, damit wird der Schanzentisch zum Nullpunkt im Koordinatensystem. Dies vereinfacht die 3D Darstellung im Frontend. Ausserdem werden weitere Daten wie Geschwindigkeit und zurückgelegte Distanz berechnet.

Werden Änderungen an Models vorgenommen, die Einfluss auf die *GraphPoints* haben, werden die betroffenen *GraphPoints* gelöscht. Beim erneuten Aufruf werden diese wieder berechnet und persistiert.

## 4.5 API Schnittstelle

Die API Schnittstellenbeschreibung befindet sich im Anhang 8.9.

## 4.5.1 Wichtige Abläufe

### Training erstellen

In Abbildung 4.4 wird die Erstellung eines neuen Trainings systemübergreifend beschrieben.

## 4.6 Prozesse und Threads

Die Web Server Gateway Interface (WSGI) Library Gunicorn (Green Unicorn), welche den Django Server über WSGI bereitstellt, verwendet Threading. In der Django Applikation werden keine Threads verwendet.

## 4.7 Continuous Integration

Es wird Continuous Integration (CI) mit Pipelines auf GitLab eingesetzt, um die Wartbarkeit des Systems zu erhöhen.

CI ermöglicht unter anderem die automatische Ausführung von Tests. Dies spart Zeit in der Entwicklung, da weniger Tests manuell durchgeführt werden müssen.[39]

### 4.7.1 Linting

Das Frontend sowie das Backend nutzen einen Linter, um die Formatierung in den Projekten einheitlich zu halten. Beim Frontend wird TS Lint und beim Backend pycodestyle verwendet. Dies ist auch in den nicht funktionalen Anforderungen vermerkt (siehe 3.5.5).

### 4.7.2 Unit Testing

Im Backend werden die Unit Tests direkt in Python ausgeführt. Im Frontend wird mit einem Headless Chrome gearbeitet, um Karma Tests auszuführen. Die End to End (E2E) Tests werden mit Protractor ebenfalls mit einem Headless Chrome gestartet.

Ausser auf den Branches Master und Production, sind fehler bei den Tests erlaubt. Dies verhindert, dass die lokale Arbeit nicht auf den Server publiziert wird, weil der Build fehlschlagen könnte. Ein Nachteil davon ist, dass bei Fehlern in Merge Requests diese aus Versehen in den Master gelangen, wo es dann zu einem Pipelinefehler kommt.

## 4.8 Continuous Deployment

Es wird Continuous Deployment mit Pipelines auf GitLab eingesetzt.

Sämtliche Commits auf dem Master-Branch werden automatisch als staging Applikation auf dem Server released.

Sämtliche Commits auf dem Production-Branch benötigen eine manuelle Interaktion auf dem GitLab Web UI, um die produktive Applikation zu ersetzen.

Jeder Branch kann als Review Applikation auf den Server deployed werden, um den aktuellen Stand unter *richtigen* Bedingungen zu testen.

### 4.8.1 Frontend

Die Webapplikation wird nach dem Build mit tar gepackt und an den Dokku Server übermittelt. Diese wird danach als static Website mit dem Nginx Reverse Proxy verknüpft. Siehe auch Appendix 8.5.3.

### 4.8.2 Backend

Das eigentliche Deployment ist ein git push auf ein Dokku-Repository auf dem Production Server von Swiss-Ski. Dieser baut, nach dem ein neuer Push eingegangen ist, die Applikation mithilfe eines Dockerfiles und verknüpft diese mit einem Nginx Reverse Proxy.

### 4.8.3 Deployment Diagramm

Das Deployment Diagramm in Abbildung 4.5 zeigt das Deployment für ein Environment.

## 4.9 Datenspeicherung

Die Speicherung der Daten wird vom OR Mapper von Django übernommen.

PostGIS wird verwendet, um Koordinaten im Raum zu verarbeiten und für den Client umzurechnen.

## 4.10 Grössen und Leistung

Die maximale Grösse der Applikation und der Daten ist abhängig vom vorhandenen Speicherplatz. Da es sich um einen Cloudserver handelt, lässt sich der Platz bei Bedarf aufrüsten.

Der Speicherplatz für Benutzer und Metainformationen zu Trainings und Sprüngen kann vernachlässigt werden.

Pro Sprung wird mit 8000 Messpunkten ( 1MB) und 2 Videos mit jeweils 10s Spieldauer ( 50MB pro Video) gerechnet.

Der Speicherbedarf für einen Sprung wird somit auf 101 MB geschätzt.

## 4.11 Lizenzen

Dieser Teil beschreibt die Runtime Lizenzen der geschriebenen Software.

### 4.11.1 Frontend

Beim Buildprozess von Angular wird eine *3rdpartylicenses.txt* Datei angelegt, welches alle relevanten Lizenzen aufführt (siehe Appendix 8.11).

In der Tabelle 4.3 werden die direkten Abhängigkeiten mit ihrer Lizenzart aufgelistet.

Library	Version	Lizenz
Angular	5.2.10	MIT
Angular Material	5.2.3	MIT
Angular CDK	5.2.3	MIT
Angular Flex-Layout	5.0.0-beta.13	MIT
Chart.js	2.7.2	MIT
chartjs-plugin-annotation	0.5.7	MIT
chartjs-plugin-zoom	0.6.3	MIT
three.js	0.91.0	MIT
zone.js	0.8.26	MIT
hammerjs	2.0.8	MIT
core-js	2.4.1	MIT
Material-Design-Icons	3.0.1	Apache-2.0
Roboto Fontface	0.9.0	Apache-2.0
RxJS	5.5.10	Apache-2.0
web-animations-js	2.3.1	Apache-2.0

Tabelle 4.3: Libraries und Lizenzen Frontend

### 4.11.2 Backend

Die Tabelle 4.4 enthält alle Libraries, die im Pipfile des Backend Projekts unter "[packages]"geführt werden.

Library	Version	Lizenz
Django Rest Framework	3.7.7	BSD
Django	2.0.3	BSD
Coreapi	2.3.3	BSD
Gunicorn	19.7.1	MIT
Django-Heroku	0.3.1	MIT
Django-Cors-Headers	2.2.0	MIT
Pillow	5.0.0	PIL License
drf-yasg	1.7.4	BSD
flex	6.13.1	BSD
swagger-spec-validator	2.1.0	Apache-2.0

Tabelle 4.4: Libraries und Lizenzen Backend

Python selbst steht unter der Python Software Foundation License (PSFL) und wird in Version 3.6.4 eingesetzt.

### 4.11.3 Erläuterung zu den Lizenzen

#### MIT

Die MIT-Lizenz ist eine permissive Lizenz, die kurz und bündig ist. Man darf mit dem Code alles tun, solange der Entwickler erwähnt und nicht haftbar gemacht wird. [40]

#### BSD

Die BSD Lizenz ist in drei Formen anzutreffen. Grundsätzlich darf die Software weiterverbreitet werden, auch in abgeänderter Form, solange die Lizenz in allen Kopien und Dokumentationen erwähnt wird. Ausserdem dürfen die Namen der Entwickler nicht für Marketingzwecke verwendet werden.

Die originale 4-Klausel-BSD-Lizenz besagt ausserdem, dass der Hersteller der Originalsoftware bei Werbungen im exakten Wortlaut erwähnt werden muss. [41]

#### Apache-2.0

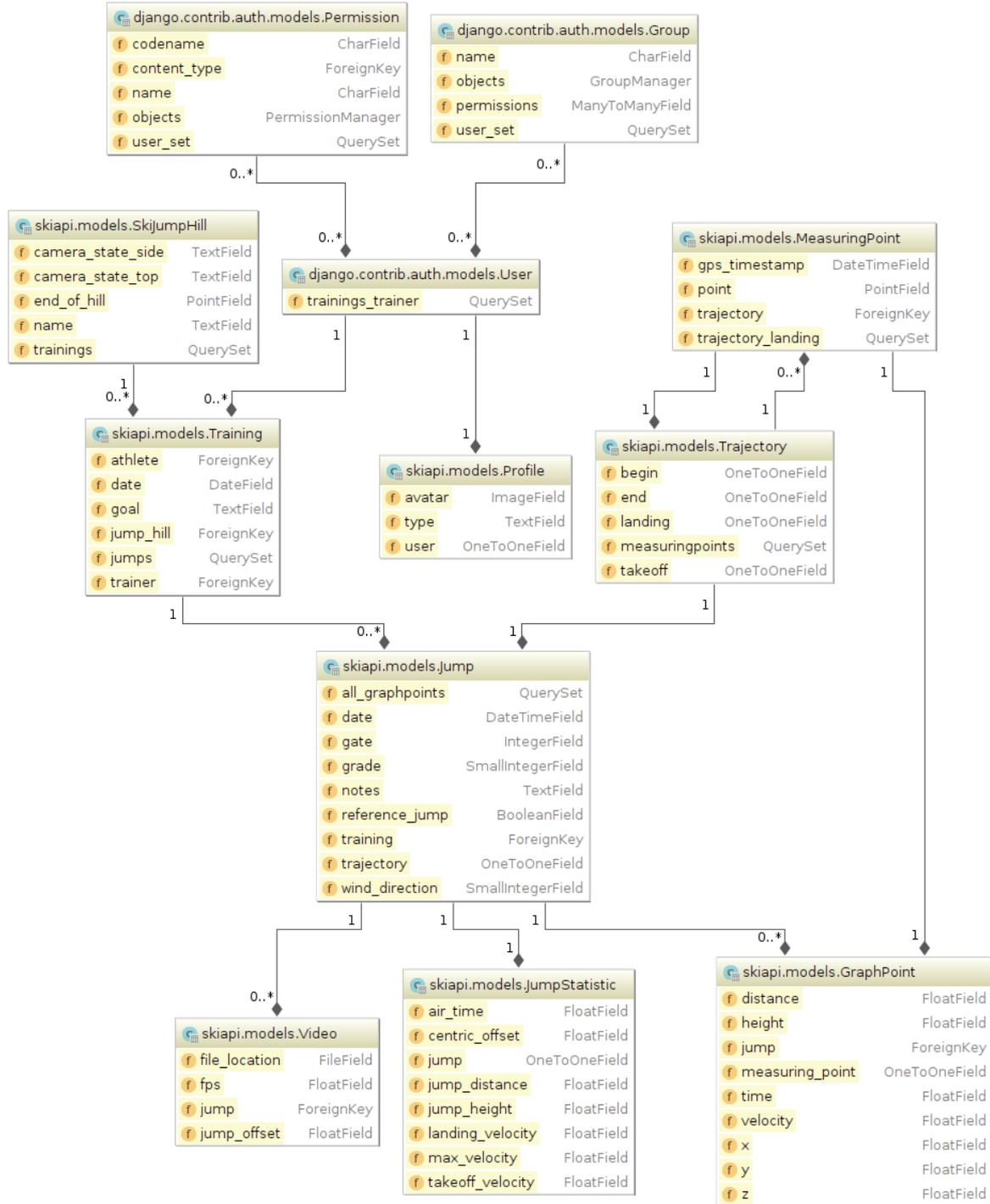
Die Apache Lizenz 2.0 ist eine permissive Lizenz ähnlich der MIT-Lizenz, bietet aber auch eine ausdrückliche Erteilung von Patentrechten von Mitwirkenden an Benutzer. [40]

#### PSFL

Die PSFL erlaubt kommerzielle Verwendung und Redistribution ohne Weitergabe des Quellcodes. [42]

#### PIL License

Eine eigene Lizenz für das Python Pillow Projekt. Es erlaubt die Verwendung und Manipulation des Projektes, solange man dessen Lizenz in allen Kopien und Dokumentationen erwähnt. In dieser Dokumentation ist sie im Abschnitt 8.12 zu finden.



Powered by yFiles

Abbildung 4.3: Django Models

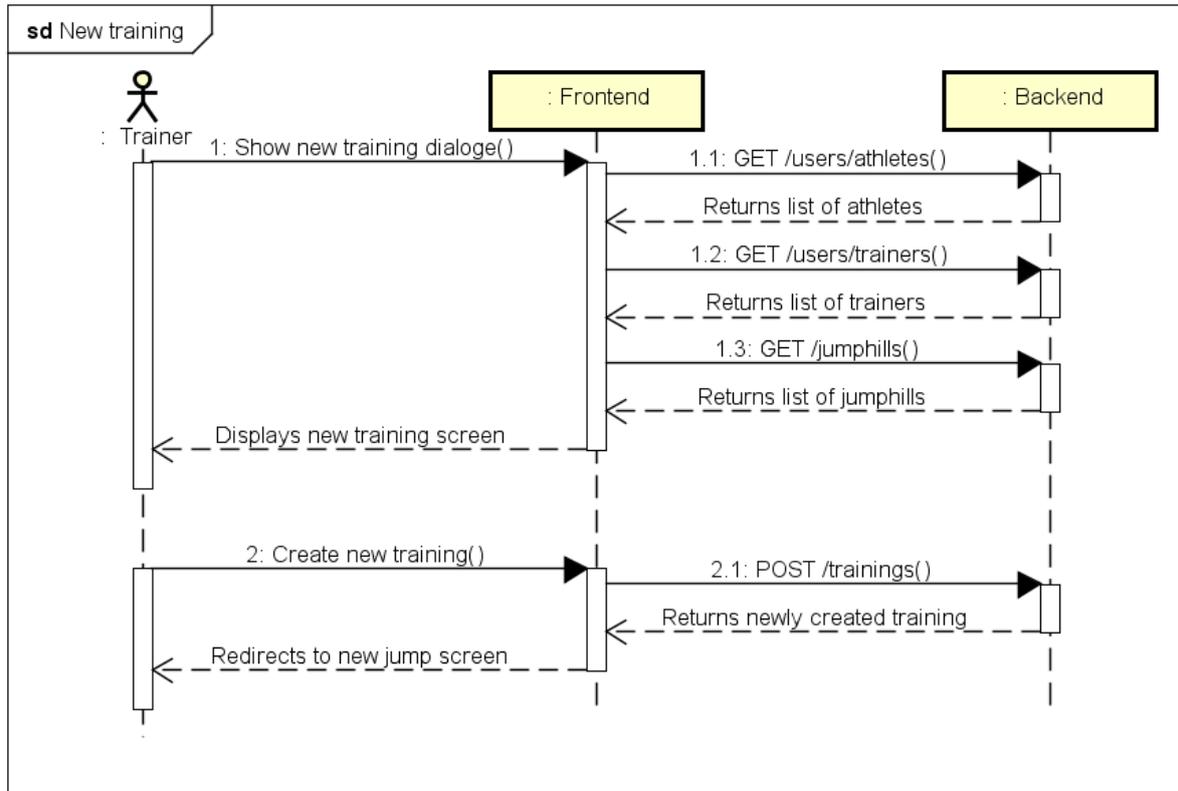


Abbildung 4.4: Neues Training erstellen SSD

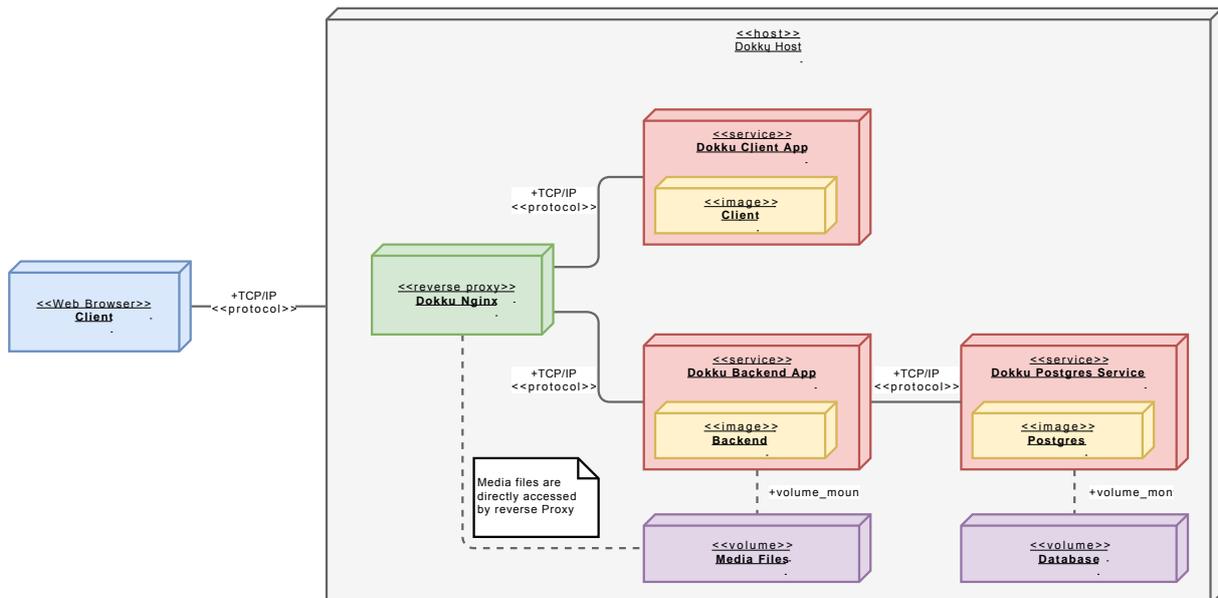


Abbildung 4.5: Deployment Diagramm

# 5 Implementation

## 5.1 Prototypen

Als erstes wurden verschiedene Prototypen des Systems implementiert. Mit diesen Prototypen konnten die technischen Risiken minimiert werden. Zudem dienten sie der Verifikation der funktionalen Anforderungen und konnten den Trainern im Trainermeeting gezeigt werden. Aus dem Feedback der Trainer konnten wieder Änderungen an den Funktionalen Anforderungen vorgenommen werden.

Die Prototypen befinden sich in Tabelle 5.1.

Prototyp	Fragestellung	Resultat
Eigene Controls für Video	Machbarkeit im Browser? Wie sieht es auf einem Tablet aus?	Slider als eigenes Control erwies sich als machbar. Funktioniert auch auf iPad.
Sprungvisualisierung (2D, 3D)	Sind mehrere Sprünge gleichzeitig darstellbar?	Dank Zooming können auch sehr nahe beieinanderliegende Trajektorien auseinandergehalten werden. Dank Einfärbung unterscheidbar.
Authentication	Wie authentisiert sich das Frontend gegenüber dem Backend?	Lösung mit Bearer Token gefunden.
Handling Sprungdaten	Muss die Trajektorie auf dem Server komprimiert werden?	Nein, dem Client können problemlos alle Punkte geschickt werden.
Aufbereitung Videodateien	Müssen die Videodateien für den Client komprimiert werden?	Nein, Client kann mit Demovideo umgehen. Keine Aufbereitung notwendig.

Tabelle 5.1: Prototypen

## 5.2 Frontend

### 5.2.1 Code Review

Am 22.05.2018 wurde ein Code Review mit Silvan Gehrig durchgeführt. Silvan unterrichtet an der HSR im Web Engineering und Design Modul das Thema Angular.

Nachfolgend sind das Reviewprotokoll sowie die daraus entstandenen Entscheidungen aufgeführt.

#### Gesamteindruck

Der Gesamteindruck ist überzeugend. Die Applikation wurde sinnvolle Module unterteilt und die Unterteilung wurde korrekt gemacht.

#### Struktur und Namen

Die Struktur ist gut gewählt. Die Module sind am richtigen Ort mit den richtigen Namen. Die Dateien haben korrekte Präfixe und sind den Richtlinien entsprechend benannt.

#### Unschönheiten

Die Navigation wurde zentral in einem Routing Module gelöst. Dies ermöglicht kein Lazy-Loading. Da der Einstiegspunkt aber das Dashboard ist, wo alle Module zusammenkommen, ist der Einfluss klein. Kommen mehr Module dazu, wird ein verteiltes Routing wichtiger. Die impliziten Weiterleitungen sind mit statischen Strings gelöst statt mit einem zentralen Routing Module. Das führt zu statischen strings verteilt über die Applikation.

Lösung: Eigener Service für Navigationen einsetzen, welcher sich um die vordefinierten Routen kümmert.

#### Mögliche Probleme

Die *forkJoin* Operation, welche mehrere Observables abwartet und dann einmal returned, kann nicht mit Fehlern umgehen. Wenn eines der gesammelten Observables einen Error wirft, wird der *forkJoin* nie beendet.

Lösungsansätze

- Eigene *saveForkJoin* Methode, welche Fehler akzeptiert.
- Die Observables mit einer *pipe* und *catchError* fehlerresistent machen, indem ein gültiges *of()* im Fehlerfall zurückgegeben wird.

Aufgrund der einfacheren Umsetzung haben wir uns für ein *catchError* entschieden, welches im Fehlerfall ein gültiges Observable generiert.

## Upgrade auf Angular 6

Gegen Ende der Bachelorarbeit ist ein Update für Angular erschienen. Silvan Gehrig wurde gefragt, ob er uns dieses Upgrade anraten würde und wie er den verbundenen Aufwand einschätze. Grundsätzlich findet er es gut, wenn man das eingesetzte Framework auf dem neusten Stand hält. Den Aufwand schätzt er klein ein, da sich wenig in fundamentales geändert habe.

Beim Versuch, das Upgrade zu machen, hat sich herausgestellt, dass RxJS die eigentliche Knacknuss darstellen wird. Da wir stark auf Observables mit RxJS setzen, ist der Aufwand, diese zu migrieren, grösser als gedacht. Das Compatibility Paket von RxJS wollen wir nicht verwenden, da es aus unserer Sicht mehr Sinn ergibt, die Applikation ganz zu migrieren und keine legacy Baustellen zu hinterlassen.

## 5.3 Backend

### 5.3.1 Code Review

Am 17. Mai 2018 wurde mit Samuel Kurath ein Code Review durchgeführt. Nachfolgend sind das Reviewprotokoll sowie die daraus entstandenen Entscheidungen aufgeführt.

#### Gesamteindruck

Sein Gesamteindruck ist positiv. Er bezeichnet seine Inputs als «jammern auf hohem Niveau».

#### Kommentare

An zwei Stellen war vor komplexen boolschen Ausdrücken ein Kommentar. Seine Empfehlung ist, diese Ausdrücke in Funktionen mit sprechenden Namen auszulagern.  
Resultat: Umgesetzt.

#### Filegrösse

Das Projekt verwendet die Standard Django File- und Ordnerstruktur. Er empfiehlt, grössere Files zu splitten. Bei Models könnte man anstatt models.py einen Ordner models erstellen, und darin jump.py, training.py etc. führen.  
Ergebnis: Aus Zeitgründen nicht umgesetzt.

#### Type Hintings

Teilweise werden Python Type Hintings verwendet. Er findet dies ungewöhnlich. PyCharm erkennt zum Teil den Typen auch anhand des Variablennamens.  
Ergebnis: Keine Änderungen notwendig.

## Abkürzungen in Unit Tests

In Unit Tests wurden Variablennamen oft abgekürzt. Variablennamen sollen ausgeschrieben werden.

Ergebnis: Variablennamen ausgeschrieben.

## Unit Testing

Kleine Methoden in Services.py nur indirekt getestet. Empfehlung ist, diese auch direkt zu testen.

Ergebnis: Durchgeführt. Fehler in der Distanzmessung entdeckt und behoben.

Weil gewisse Helferfunktionen in einem Unterordner abgelegt sind, musste der Python-path manuell im Test angepasst werden. Gemeinsam fanden wir eine Lösung, bei der der Path nicht mehr manipuliert werden muss.

## Datenklassen

In Python können anstatt einer Datenklasse Dictionarys oder Namedtuples verwendet werden.

Ergebnis: Aus Zeitgründen nicht umgesetzt.

## Fehlerbehandlung

Bei der Funktion `textitcaclulate_one_graphpoint` kann eine Division durch Null stattfinden, wenn zwei aufeinanderfolgende Messpunkte den gleichen Zeitstempel haben. Verbesserungsvorschläge:

- Exception
- Loggen
- Graceful behandeln (Geschwindigkeit auf Null setzen)

Die letzte Variante wurde umgesetzt.

## 5.4 Code Metriken

Die Code Metriken befinden sich in der Tabelle 5.2. Diese Werte wurden mit SonarQube [43] berechnet.

Der Code im Backend weist eine eher geringe zyklomatische Komplexität auf. Dies liegt daran, dass viele Entscheidungen durch das Framework über Annotationen getätigt werden. Im Frontend hingegen werden Elemente abhängig von den vorhandenen Daten angezeigt (Data-Binding). Diese Logik wird in unserem Code behandelt. Die grösste Komplexität haben die beiden Directives, welche die 2D und 3D Visualisierung der Messpunkte ermöglichen.

Metrik	Frontend	Backend
Lines of Code	5610	2538
Functions	558	164
Classes	92	98
Cyclomatic Complexity	908	249
Code Coverage	70.72%	92%

Tabelle 5.2: Code Metriken

Code Coverage Frontend befindet sich in Listing 5.1, Backend in Listing 5.2.

```

1 > ng test --watch=false --code-coverage
2 [...]
3 ===== Coverage summary =====
4 Statements   : 70.72% ( 1476/2087 )
5 Branches    : 23.64% ( 100/423 )
6 Functions    : 61.24% ( 346/565 )
7 Lines       : 69.65% ( 1301/1868 )
8 =====
  
```

Listing 5.1: Frontend Test Coverage

Name	Stmts	Miss	Cover
-----			
skiapi/__init__.py	0	0	100%
skiapi/admin.py	73	21	71%
skiapi/apps.py	5	0	100%
skiapi/models.py	114	18	84%
skiapi/permissions.py	26	1	96%
skiapi/serializers.py	138	4	97%
skiapi/services.py	97	0	100%
skiapi/signals.py	27	0	100%
skiapi/swagger.py	12	0	100%
skiapi/tests/__init__.py	0	0	100%
skiapi/tests/confptest.py	12	0	100%
skiapi/tests/helpers/__init__.py	0	0	100%
skiapi/tests/helpers/factories.py	45	0	100%
skiapi/tests/helpers/helpers.py	30	2	93%
skiapi/urls.py	14	0	100%
skiapi/views.py	139	8	94%
skijumpapi/__init__.py	0	0	100%
skijumpapi/settings.py	35	1	97%
skijumpapi/urls.py	8	1	88%
skijumpapi/wsgi.py	4	4	0%
-----			
TOTAL	779	60	92%

Listing 5.2: Test Coverage Python

## 5.5 Security Review

Gemeinsam haben die beiden Autoren das Front- und Backend am 05. Juni 2018 auf die OWASP Top 10 Lücken [44] geprüft.

### A1:2017-Injection

In Backend als User ' versucht, einzuloggen. Das Login wurde nicht akzeptiert. Gemäss Django Security Guide [45] werden parametrized SQL Queries verwendet. In unserem Code werden keine händischen SQL Queries ausgeführt.

### A2:2017-Broken Authentication

Das Backend forciert minimale Passwort Anforderungen:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Das Frontend sendet bei jedem Request ein Token mit, welches es vom Backend nach dem erfolgreichen Login erhalten hat. Dieses Token wird pro User in der Datenbank gespeichert. Ein Logout löscht dieses Token einzig aus dem Session Storage des Browsers.

Passwörter werden mit dem PBKDF2 Algorithmus gehasht.

Die Applikation bietet keinen Schutz vor Brute Force attacken. Sie hat keine 2-Faktor-Authentisierung. Dieses Verhalten wurde am 11.06.2018 dem Business Partner kommuniziert.

### A3:2017-Sensitive Data Exposure

Der Transport der übertragenen Daten ist TLS verschlüsselt. Zum Prüfen der SSL Einstellungen wurde ein Test bei sslabs [46] durchgeführt. Das Resultat ist Grade A.

Verbesserungsmöglichkeiten:

- DNS CAA Record [47]
- Weak Ciphers entfernen (z.B. Vorschlag von Mozilla implementieren [48])
- Strict Transport Security Header verwenden

### A4:2017-XML External Entities (XXE)

Die Applikation verwendet kein XML im Backend. Daher wird diese Kategorie nicht geprüft.

### A5:2017-Broken Access Control

Videodateien sind ohne Authentisierung aufrufbar. Athleten können über die API Felder sehen, die sie nicht sehen dürften (Grade, Sprungnotizen). Bei der Implementierung wurde primär darauf geachtet, dass nicht angemeldete User keine Trainingsdaten sehen. Dieses Verhalten wurde am 11.06.2018 dem Business Partner kommuniziert.

### A6:2017-Security Misconfiguration

Das *DEBUG* Runtime Flag ist einzig in der Development Umgebung eingeschaltet. Diese ist jedoch öffentlich zugreifbar.

Empfehlung: Zugriff auf Development und Staging Systeme einschränken. Dies könnte zum Beispiel über einen SSH Tunnel umgesetzt werden.

Der Videofileupload erlaubt das Hochladen von beliebigen Dateitypen für authentifizierte User. Da der Zugriff auf die Dateien direkt über den Eintrittsserver gehen, sind diese Files öffentlich verfügbar.

Empfehlung: Authentisierung des Filezugriffs implementieren. Im Eintrittsserver den Content-Type statisch auf *video/mp4* setzen.

### A7:2017-Cross-Site-Scripting (XSS)

In Backend einem Trainer und einem Athleten den Namen `<script>alert('hoi')</script>` geben. Einem Sprung und einem Training diesen Text als Notiz hinzugefügt.

Resultat: Weder im Admin UI noch in der Angular Applikation wurden Javascript Alerts ausgeführt.

### A8:2017-Insecure Deserialization

Es wird ausschliesslich JSON zur Serialisierung verwendet. Es werden keine binären Objekte serialisiert. Daher wird diese Kategorie nicht geprüft.

### A9:2017-Using Components with Known Vulnerabilities

Im Backend lassen wurde der Pipenv dependency checker (siehe Listing 5.3) laufen gelassen. Es wurden keine Vulnerabilities gefunden.

```
1 $ pipenv check
2 Checking PEP 508 ...requirements
3 Passed!
4 Checking installed package ...safety
5 All good!
```

Listing 5.3: Pipenv dependency check

Im Frontend wird npm audit verwendet, siehe Listing 5.4. Der vollständige Scan befindet sich auf Seite 128 im Appendix 8.25.

```
1 $npm audit
2 [...]
3 found 19 vulnerabilities (1 low, 17 moderate, 1 high) in 7615 scanned packages
4 run `npm audit fix` to fix 13 of them.
5 6 vulnerabilities require manual review. See the full report for details.
```

Listing 5.4: NPM dependency check

Die gefundenen Lücken betreffen ausschliesslich Unterabhängigkeiten von Protractor und Karma. Diese Dependencies werden nur für Entwicklungszwecke benötigt und sind in der Webseite selbst nicht inkludiert.

Empfehlung: Auf aktualisierte Versionen der Dependencies updaten.

### **A10:2017-Insufficient Logging & Monitoring**

Die Applikation verwendet standard Django Logging. Monitoring der Serverumgebung ist nicht Teil dieser Arbeit.

## **5.6 User Acceptance Test**

Am 31.05.2018 wurde mit dem Trainer Martin Künzle von Swiss-Ski bei einem Training vor Ort ein User Acceptance Test (UAT) dieses Produktes durchgeführt.

### **5.6.1 Vorbereitung**

Der Test wird auf einem Entwicklerlaptop durchgeführt.

Der Proband erhält die Aufgaben schriftlich, die er mit der Applikation lösen soll.

Die Aufgaben müssen in der vorgegebenen Reihenfolge gelöst werden. Falls ein User die Aufgabe nicht lösen kann, wird das Vermerkt und mit Hinweisen sichergestellt, dass mit den nachfolgenden Aufgaben weitergemacht werden kann.

#### **Hilfsmittel**

- Zugangsdaten (Username, Passwort, URL)
- USB Stick mit Videodatei (bei entsprechender Aufgabe)
- Vorab hochgeladene Messreihe mit aktuellem Datum

### **5.6.2 Szenario**

Stellen Sie sich vor, Sie sind Franz Meier, Trainer bei Swiss-Ski. Sie trainieren heute mit Ihrem Athleten Christian Schmidt auf der Grossschanze Einsiedeln mit HS117. Es soll die neue Schuhbindung getestet werden.

Die Sprünge von Christian Schmidt wollen Sie dieses Mal im neuen GPS-System dokumentieren. Hierzu haben Sie die folgenden Daten in einer E-Mail erhalten:

## Zugangsdaten

Applikation: <https://testing.gps.skispringen.ch/>

Username: franzmeier

Passwort: xxx

## Hintergrund

Wir nehmen an, dass Sie das GPS System aktiviert hatten und Christian Schmidt seinen ersten Sprung gerade absolviert hat. Er hat dabei den Helm mit dem GPS getragen. Es war ein guter Sprung ab Gate 28 bei relativ viel Gegenwind. Sie haben diesen Sprung mit Ihrer Video-Kamera aufgezeichnet und auf ihren Computer geladen.

## Aufgaben

Die Aufgaben und Resultate sind in der Tabelle 5.3 aufgeführt.

### Nummer

### Aufgabe

1

Nutzen Sie das System, um den Sprung mit den Messdaten und weiteren wichtigen Daten (inkl. Video) zu dokumentieren.

### Bemerkungen

- Login einwandfrei
- Im Dashboard ist es dem User nicht klar, dass es sich um einen alten Sprung handelt
- User fügt Video zu altem Sprung hinzu (Unglück. Da dieser Sprung weder ein Video noch eine Messreihe hatte, sah es aus wie ein neuer Sprung)
- Video hochgeladen, ohne es am Absprung anzuhalten. Absprungpunkt war ganz am Ende des Videos gesetzt.
  - Er hat selber gemerkt, was er falsch gemacht hatte
  - Konnte das Video löschen und mit korrektem Absprungszeitpunkt neu hinzufügen
- Auswahl des Absprung- und Landepunktes aus der Messreihe intuitiv richtig gewählt, da er weiss, wie sich die Geschwindigkeiten entwickeln.
  - Für ungeschulte Person ist dies nicht ersichtlich

<b>Nummer</b>	<b>Aufgabe</b>
<b>2</b>	Stellen Sie sicher, dass das Video abgespielt werden kann und die Messreihe sich mit dem Video deckt.
Bemerkungen	Klickte intuitiv auf das Video, womit dieses abgespielt wurde. Dies könnte bei anderen Probanden nicht direkt erkannt werden.
<b>3</b>	Sie haben vom Kollegen gehört, dass man mit dem System auch Geschwindigkeiten der Athleten beim Sprung analysieren kann. Finden Sie heraus, wie schnell Christian Schmidt beim Anlauf, 3 Sekunden vor Absprung war.
Bemerkungen	Konnte die Stelle (-3s) finden und die Geschwindigkeit korrekt ablesen. Zuerst versuchte er den Wert aus der Grafik abzulesen, bevor er gesehen hat, dass es auf der linken Seite steht.
<b>4</b>	Nehmen wir an, dass nun schon mehrere Sprünge absolviert wurden. Vergleichen Sie nun den soeben erfassten Sprung mit dem Sprung 5. Bei welchem Absprung hatte Christian Schmidt die höhere Geschwindigkeit?
Bemerkungen	<ul style="list-style-type: none"><li>• Anfangs etwas unsicher beim Erkennen der Sprünge. Aktueller und Vergleichssprung.</li><li>• Wählte alle Referenz Sprünge an, da es genau zwei zur Auswahl hatte. Vergleich somit drei Sprünge.</li><li>• Deselektierte die falschen Sprünge wieder.</li><li>• Vergleich hat geklappt, Geschwindigkeit konnte korrekt abgelesen werden.</li></ul>
<b>5</b>	Ihre Erkenntnis aus den Sprüngen ist, dass die Bindung sehr geeignet ist und weiter getestet werden soll. Hinterlegen, Sie die entsprechende Information so, dass Sie diese auch später noch finden können.

**Nummer      Aufgabe**

Bemerkungen

- Es war nicht klar, wo diese Information abgelegt werden soll. Sprung oder Training standen zur Auswahl.
- Er entschied sich dann für den Sprung und somit für das in diesem Fall vorgesehene Notizfeld.
- Anregung des Users: Anzeigen der Sprungnotiz in der Übersicht von den Trainings für Wiedererkennung eines Sprungs.

Tabelle 5.3: UAT Resultate

**Erkenntnisse**

Aufgrund der Schwierigkeiten im Programmeinstieg, welche bei Aufgabe 1 zu sehen sind, schlagen wir als Erweiterung ein separates Einstiegs-Dashboard vor, welches die möglichen Aktionen übersichtlich zeigt (siehe Tabelle 5.4).

<b>Aktion</b>	<b>URL</b>
Neues Training hinzufügen	/trainings/add-training
Neuer Sprung zum letzten Training hinzufügen	/(latest-training-id)/add-jump
Vergangene Trainings anzeigen	/trainings
Aktuellsten Sprung öffnen	/jumps/(latest-jump-id)/dashboard

Tabelle 5.4: Dashboard Aktionen

Insgesamt zeigte der UAT, dass die Applikation nach einer kurzen Schulung durch die Trainer bedient werden kann.

## 6 Ausblick

Beim Trainingsbesuch vom 31. Mai 2018 wurde die Applikation erstmals für ein Training eingesetzt. Bis dahin wurde ausschliesslich mit einer Messung und Mock Daten gearbeitet. Um die Benutzererfahrung noch besser zu machen, werden folgende Erweiterungen der Lösung empfohlen. Diese Liste wurde gemeinsam mit Samuel Kurath erarbeitet.

### 6.1 Hohe Priorität

- Umsetzen der Erkenntnisse aus dem Security Review
- Testen des ganzen Ablaufes und daraus gewonnene Erkenntnisse umsetzen. (Sprünge messen, Training erfassen, Trainingsanalyse, ...)
- Automatische Perspektiven für 3D Messungen erstellen (Transformation zwischen ECEF- und ENU-Koordinaten)
- 3D Modell der Schanze in Backend erfassen können
- Erfassen neuer Schanzen (Absprungpunkt, Kamerapositionen)

### 6.2 Mittlere Priorität

- Absprung, Landung, Start und Ende automatisch im Backend erkennen
- Usability verbessern, Play Button um Video abzuspielen hinzufügen
- Rückmeldung bei Formularfehlern (Umgang mit 4xx Errors von der API)

### 6.3 Geringe Priorität

- Export der Daten als CSV
- Admin Tasks in Frontend durchführen können
- Videobeginn/end auf Topbar (Stepper) anzeigen
- Automatisierte zuweisung einer Messreihe an den Sprung
- Sprungvergleich 3D: Position im Raum bei allen Sprüngen anzeigen
- Videozugriff Authentisieren

# 7 Verzeichnisse

## 7.1 Glossar und Abkürzungen

### **certbot**

Certbot ist eine Software, die automatisiert bei Let's encrypt Zertifikate bezieht. [49]

### **Chart.js**

Chart.js ist eine Javascript Graph Library [23]

### **CI**

Continuous Integration

### **Confluence**

Dokumentenmanagementsystem von Atlassian [50]

### **DBMS**

Datenbank Management System

### **Debian**

Eine Linux Distribution.

### **Django**

Eine Library für Python, die das Erstellen von dynamischen Webseiten vereinfacht.

### **Docker**

Docker ist ein Wrapper um Linux-Kernel Prozessvirtualisierungsfeatures und vereinfacht die Automatisierung von Applikationsdeployments. [51]

### **Dokku**

Ein PaaS mit Docker welches das Bauen und Verwalten von Applikationen vereinfacht. [52]

### **DRF**

Django Rest Framework

## **E2E**

End to End

## **Jira**

Jira ist ein webbasierendes Projektmanagement Tool von Atlassian. [53]

## **Karma**

Test-runner für JavaScript Projekte

## **Let's encrypt**

Let's encrypt ist eine CA, die über eine API kostenlose Serverzertifikate herausgibt. [54]

## **Nginx**

Nginx (Engine-X) ist ein Webserver und Reverse-Proxy. [55]

## **Node.js**

JavaScript Runtime basierend auf Chromes V8

## **NPM**

Node Package Manager

## **OpenAPI**

Standard für API Dokumentation

## **PaaS**

Platform as a Service

## **PBKDF2**

Password-Based Key Derivation Function 2, eine standardfunktion um Passwörter sicher zu speichern

## **PostGIS**

PostGIS ist eine Geo-Datenbank erweiterungen für PostgreSQL

## **PostgreSQL**

PostgreSQL ist ein freies Datenbank Management System (DBMS). [35]

## **PSFL**

Python Software Foundation License

## **PyCharm**

Eine Python IDE von JetBrains.

## **Sim Daltonism**

macOS und iOS Applikation zum Simulieren von Farbenblindheit. Von Michel Fortin (michelf.ca/projects/sim-daltonism)

## **Slack**

Slack ist ein Instant Messaging Tool [56]

## **SPA**

Single Page Application

## **SQLite**

SQLite ist ein DBMS, welches lokal läuft.

## **SVG**

Scalable Vector Graphics

## **Swagger.io**

API Dokumentations Tool

## **three.js**

three.js ist eine Javascript 3D Library [15]

## **TypeScript**

TypeScript erlaubt das schreiben von typisierten JavaScript Code

## **UAT**

User Acceptance Test

## **VPS**

Virtual Private Server

## **WebStorm**

Eine HTML, JavaScript IDE von JetBrains.

## **WSGI**

Web Server Gateway Interface

## **Yarn**

Package Manager für Node Module, alternative zu NPM

## 7.2 Abbildungsverzeichnis

0.1	Eindrücke während der Bachelorarbeit . . . . .	5
3.1	Kontext Diagramm . . . . .	14
3.2	Use Case Diagramm (Administrativ) . . . . .	15
3.3	Use Case Diagramm . . . . .	16
3.4	Domainmodell . . . . .	18
3.5	UI Entwürfe aus Balsamiq für Confluence . . . . .	19
4.1	Systemübersicht . . . . .	23
4.2	Environments . . . . .	30
4.3	Django Models . . . . .	38
4.4	Neues Training erstellen SSD . . . . .	39
4.5	Deployment Diagramm . . . . .	39
8.1	Simulierte Farbenblindheit . . . . .	62
8.2	iPad Screenshot Dashboard . . . . .	63
8.3	iPad Screenshot Trainings . . . . .	63
8.4	iPad Screenshot 3D Sprungvergleich . . . . .	64
8.5	Aufgeklapptes Training in der Trainingsübersicht . . . . .	91
8.6	Der Action Button befindet sich unten rechts auf dem Bildschirm . . . . .	91
8.7	Sprung bearbeiten mit Trajektorie zum Löschen markiert . . . . .	92
8.8	Admin UI Startseite . . . . .	94
8.9	Admin UI Users . . . . .	94
8.10	Kamera State . . . . .	96

## 7.3 Tabellenverzeichnis

4.1	Libraries im Frontend . . . . .	26
4.2	Libraries im Backend . . . . .	28
4.3	Libraries und Lizenzen Frontend . . . . .	36
4.4	Libraries und Lizenzen Backend . . . . .	36
5.1	Prototypen . . . . .	40
5.2	Code Metriken . . . . .	44
5.3	UAT Resultate . . . . .	50
5.4	Dashboard Aktionen . . . . .	50
8.1	Resultate . . . . .	61
8.2	Git Repositories . . . . .	87

## 7.4 Literatur

- [1] *DGNSS Messsystem für Skispringer*, [Online; aufgerufen am 07. Juni 2018]. Adresse: <https://www.hsr.ch/de/studium/bachelor/studiengaenge/elektrotechnik-e/studieninhalt/studien-und-bachelor-arbeiten/seminarpaper/dgnss-messsystem-fuer-skispringer/>.
- [2] *Balsamiq Wireframes - Confluence Server*, [Online; aufgerufen am 13. Juni 2018]. Adresse: <https://marketplace.atlassian.com/apps/256/balsamiq-wireframes-confluence-server?hosting=server&tab=overview>.
- [3] *Angular - Style Guide*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://angular.io/guide/styleguide/>.
- [4] *TSLint - An extensible linter for the TypeScript language*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://palantir.github.io/tslint/>.
- [5] *pycodestyle (formerly called pep8) - Python style guide checker*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://github.com/pycqa/pycodestyle/>.
- [6] *Password management in Django*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://docs.djangoproject.com/en/2.0/topics/auth/passwords/>.
- [7] *Angular - One framework. Mobile & desktop*. [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://angular.io/>.
- [8] *Angular Material - Material Design components for Angular*. [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://material.angular.io/>.
- [9] *Angular Docs*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://angular.io/docs/>.
- [10] *TypeScript Manual - Angular*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://www.typescriptlang.org/docs/handbook/angular.html>.
- [11] *React - A JavaScript library for building user interfaces*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://reactjs.org/>.
- [12] *MATERIAL-UI - React components that implement Google's Material Design*. [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://material-ui.com/>.
- [13] *TypeScript Manual - React & Webpack*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://www.typescriptlang.org/docs/handbook/react-&-webpack.html>.
- [14] *React - Static Type Checking*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://reactjs.org/docs/static-type-checking.html>.
- [15] *three.js - Javascript 3D library*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://threejs.org>.
- [16] *three.js docs - OrthographicCamera*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://threejs.org/docs/index.html#api/cameras/OrthographicCamera>.

- [17] *three.js docs - Drawing lines*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://threejs.org/docs/index.html#manual/introduction/Drawing-lines>.
- [18] *three.js docs - SphereGeometry*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://threejs.org/docs/index.html#api/geometries/SphereGeometry>.
- [19] *vis.js - A dynamic, browser based visualization library*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <http://visjs.org>.
- [20] *vis.js - Configuration Options*, [Online; aufgerufen am 05. Juni 2018]. Adresse: [http://visjs.org/docs/graph3d/#Configuration\\_Options](http://visjs.org/docs/graph3d/#Configuration_Options).
- [21] *vis.js - Configuration Options*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <http://visjs.org/docs/graph3d/#Events>.
- [22] *vis.js - Data Fromat*, [Online; aufgerufen am 05. Juni 2018]. Adresse: [http://visjs.org/docs/graph3d/#Data\\_Format](http://visjs.org/docs/graph3d/#Data_Format).
- [23] *Chart.js - Simple yet flexible JavaScript charting for designers & developers*, [Online; aufgerufen am 05. Juni 2018]. Adresse: [http://visjs.org/docs/graph3d/#Data\\_Format](http://visjs.org/docs/graph3d/#Data_Format).
- [24] *Chart.js - Line*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <http://www.chartjs.org/docs/latest/charts/line.html>.
- [25] *Zoom and pan plugin for Chart.js*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://github.com/chartjs/chartjs-plugin-zoom>.
- [26] *Annotation plugin for Chart.js*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://github.com/chartjs/chartjs-plugin-annotation>.
- [27] *vis.js - Graph2d*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <http://visjs.org/docs/graph2d>.
- [28] M. Krämer und R. Gutbell, „A Case Study on 3D Geospatial Applications in the Web Using State-of-the-art WebGL Frameworks“, in *Proceedings of the 20th International Conference on 3D Web Technology*, Ser. Web3D '15, Heraklion, Crete, Greece: ACM, 2015, S. 189–197, ISBN: 978-1-4503-3647-5. DOI: 10.1145/2775292.2775303. Adresse: <http://doi.acm.org/10.1145/2775292.2775303>.
- [29] *Node.js*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://nodejs.org/en/>.
- [30] *Express - Node.js web application framework*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://expressjs.com/>.
- [31] *The Web framework for perfectionists with deadlines / Django*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://www.djangoproject.com/>.
- [32] *Django REST framework*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://www.django-rest-framework.org/>.
- [33] *MariaDB.org - Supporting continuity and open collaboration*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://mariadb.org>.

- [34] *GeoDjango Installation*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://docs.djangoproject.com/en/2.0/ref/contrib/gis/install/#spatial-database>.
- [35] *PostgreSQL*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://www.postgresql.org>.
- [36] *LEGRIA HF G40*, [Online; aufgerufen am 05. Juni 2018]. Adresse: [https://de.canon.ch/support/consumer\\_products/products/camcorders/digital/hdv\\_series/legria\\_hf\\_g40.aspx?type=faq&search=1&gsa\\_support\\_product=LEGRIA%20HF%20G40&q=gps&searchurl=/support/consumer\\_products/content/faq/?itemid=tcm:88-1340115](https://de.canon.ch/support/consumer_products/products/camcorders/digital/hdv_series/legria_hf_g40.aspx?type=faq&search=1&gsa_support_product=LEGRIA%20HF%20G40&q=gps&searchurl=/support/consumer_products/content/faq/?itemid=tcm:88-1340115).
- [37] *LEGRIA HF G26*, [Online; aufgerufen am 05. Juni 2018]. Adresse: [https://de.canon.ch/support/consumer\\_products/products/camcorders/digital/hdv\\_series/legria\\_hf\\_g26.aspx?type=faq&search=1&gsa\\_support\\_product=LEGRIA%20HF%20G26&q=gps&searchurl=/support/consumer\\_products/content/faq/?itemid=tcm:88-1600132](https://de.canon.ch/support/consumer_products/products/camcorders/digital/hdv_series/legria_hf_g26.aspx?type=faq&search=1&gsa_support_product=LEGRIA%20HF%20G26&q=gps&searchurl=/support/consumer_products/content/faq/?itemid=tcm:88-1600132).
- [38] *LEGRIA GX 10*, [Online; aufgerufen am 05. Juni 2018]. Adresse: [https://de.canon.ch/support/consumer\\_products/products/camcorders/digital/hdv\\_series/legria\\_gx10.aspx?type=faq&search=1&gsa\\_support\\_product=LEGRIA%20GX10&q=gps&searchurl=/support/consumer\\_products/content/faq/?itemid=tcm:88-1610993](https://de.canon.ch/support/consumer_products/products/camcorders/digital/hdv_series/legria_gx10.aspx?type=faq&search=1&gsa_support_product=LEGRIA%20GX10&q=gps&searchurl=/support/consumer_products/content/faq/?itemid=tcm:88-1610993).
- [39] R. Tim, S. Tanachutiwat, M. Vukadinovic, H. J. Schlebusch und H. Lichter, „Continuous integration processes for modern client-side web applications“, in *2017 International Electrical Engineering Congress (iEECON)*, März 2017, S. 1–4. DOI: [10.1109/IEECON.2017.8075805](https://doi.org/10.1109/IEECON.2017.8075805).
- [40] *Choose an open source licence*, [Online; aufgerufen am 30. Mai 2018]. Adresse: <https://choosealicense.com/>.
- [41] *GNU - Das BSD-Lizenzproblem*, [Online; aufgerufen am 30. Mai 2018]. Adresse: <https://www.gnu.org/licenses/bsd.de.html>.
- [42] *PSFL - Python Software Foundation Lizenz*, [Online; aufgerufen am 5. Juni 2018]. Adresse: <https://docs.python.org/3/license.html#psf-license-agreement-for-python-release>.
- [43] *SonarQube - Continous Code Quality*, [Online; aufgerufen am 13. Juni 2018]. Adresse: <https://www.sonarqube.org/>.
- [44] *OWASP Top 10 - 2017*, [Online; aufgerufen am 05. Juni 2018]. Adresse: [https://www.owasp.org/images/7/72/OWASP\\_Top\\_10-2017\\_%28en%29.pdf.pdf](https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf).
- [45] *Security in Django*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://docs.djangoproject.com/en/2.0/topics/security/>.
- [46] *SSL Report: api-dev.gps.skispringen.ch*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://www.ssllabs.com/ssltest/analyze.html?d=api%2ddev.gps.skispringen.ch&hideResults=on&latest>.

- [47] *CAA Mandated by CA/Browser Forum*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://blog.qualys.com/ssllabs/2017/03/13/caa-mandated-by-cabrowser-forum>.
- [48] *Mozilla - Server Side TLS*, [Online; aufgerufen am 05. Juni 2018]. Adresse: [https://wiki.mozilla.org/Security/Server\\_Side\\_TLS](https://wiki.mozilla.org/Security/Server_Side_TLS).
- [49] *Certbot*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://certbot.eff.org>.
- [50] *Confluence - Team Collaboration Software*, [Online; aufgerufen am 13. Juni 2018]. Adresse: <https://www.atlassian.com/software/confluence>.
- [51] *Docker Project*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://www.docker.com>.
- [52] *Dokku - The smallest PaaS implementation you've ever seen*, [Online; aufgerufen am 30. Mai 2018]. Adresse: <http://dokku.viewdocs.io/dokku/>.
- [53] *Jira*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://www.atlassian.com/software/jira>.
- [54] *Let's Encrypt*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://letsencrypt.org>.
- [55] *Nginx*, [Online; aufgerufen am 05. Juni 2018]. Adresse: <https://nginx.org>.
- [56] *Slack*, [Online; aufgerufen am 07. Juni 2018]. Adresse: <https://slack.com/intl/de-de/features>.
- [57] *Dokku - dokku-letsencrypt (Beta)*, [Online; aufgerufen am 30. Mai 2018]. Adresse: <https://github.com/dokku/dokku-letsencrypt>.
- [58] *Angular - Material Design Documents*, [Online; aufgerufen am 13. Juni 2018]. Adresse: <https://github.com/angular/material2/wiki/Design-doc-directory/>.
- [59] *Django - Coding style*, [Online; aufgerufen am 06. Juni 2018]. Adresse: <https://docs.djangoproject.com/en/2.0/internals/contributing/writing-code/coding-style/>.
- [60] *Install GitLab Runner using the official GitLab repositories*, [Online; aufgerufen am 5. Juni 2018]. Adresse: <https://docs.gitlab.com/runner/install/linux-repository.html>.
- [61] *Dokku - Getting Started with Dokku*, [Online; aufgerufen am 30. Mai 2018]. Adresse: <http://dokku.viewdocs.io/dokku~v0.11.6/getting-started/installation/>.
- [62] *dokku postgres*, [Online; aufgerufen am 30. Mai 2018]. Adresse: <https://github.com/dokku/dokku-postgres>.
- [63] *Angular - Quick Start*, [Online; aufgerufen am 13. Juni 2018]. Adresse: <https://v5.angular.io/guide/quickstart/>.

# 8 Appendix

## 8.1 Test Zielerreichung

Am 6. Juni haben die Autoren die Zielerreichung getestet.

Typ	Anforderung	Test	Resultat
FA	Visualisierung Sprungdaten	UAT	OK
FA	Vergleich Sprungdaten	UAT	OK
FA	Trainingsverwaltung	UAT	OK
FA	Videoupload	UAT	OK
FA	Reagieren auf neue Messdaten	UAT	OK
A	Applikation reagiert auf Messdaten und nimmt diese in Datenbank auf	Trainingsbesuch	OK
A	Zusätzliche Angaben auf Sprüngen speichern	UAT	OK
NFA	Farben für Sehbehinderte in Ordnung	Siehe 8.1	OK
NFA	Material Designrichtlinien eingehalten	Siehe 8.1	OK
NFA	Unterstützte Plattformen getestet	Siehe 8.1	OK
NFA	Maximal 2 Minuten Datenimportdauer	Trainingsbesuch	OK
NFA	Einhaltung Angular Guidelines	Siehe 8.1	OK
NFA	Keine TSLint Fehler	Siehe 8.1	OK
NFA	Einhaltung Django Guidelines	Siehe 8.1	
NFA	Keine pycodestyle Fehler	Siehe 8.1	OK
NFA	3 E2E Tests vorhanden	Siehe 8.1	Partially OK
NFA	Wichtigste Frontend Komponenten mit Unit Tests geprüft	Siehe 8.1	OK
NFA	Wichtigste Backend Komponenten mit Unit Tests geprüft	Siehe 8.1	OK
NFA	Reivews bei Merge Requests	Siehe 8.1	OK
NFA	Manuals vorhanden	Siehe 8.7 und 8.6	OK
NFA	Datenbank Backup über Postgres funktioniert	Siehe 8.5.4	OK
NFA	Anleitung Backup und Restore vorhanden	Siehe 8.5.4	OK
NFA	Security Test vorhanden	Siehe 5.5	OK
NFA	Passwörter werden mit PBKDF2 gespeichert	Siehe [6]	OK
NFA	Berechtigungen können an Athleten gegeben und entzogen werden	Siehe 8.1	Partially OK
NFA	Sprung kann nur mit entsprechender Berechtigung angezeigt werden	Siehe 8.1	OK
NFA	Transportverschlüsselung vorhanden	Siehe [57]	OK

Typ	Anforderung	Test	Resultat
NFA	Google Chrome 67 wird unterstützt	Siehe 8.1	OK
NFA	iOS mit Safari auf iPad Pro wird unterstützt	Siehe 8.1	Partially OK
NFA	Fehler werden verständlich dargestellt	Siehe 8.1	Partially OK

Tabelle 8.1: Resultate

### Farbenfehlsichtigkeit

Die Farbauswahl in der Applikation wurde mit Sim Daltonism auf einem Macbook getestet. Es wurde darauf geachtet, dass die ersten vier Farben, welche für die Darstellung beim Sprungvergleich verwendet werden, auch mit einer Farbsehschwäche unterschieden werden können.

In der Abbildung 8.1 ist der Sprungvergleich mit verschiedenen Sehschwächen simuliert dargestellt. Auffallend ist die grün / rote Darstellung in der Sidebar für die Höhendifferenz. Man erkennt deutlich, dass diese farblich nicht mehr zu unterscheiden sind. Da diese aber mit einem Plus oder Minus Zeichen angezeigt werden, sind sie trotzdem gut zu unterscheiden.

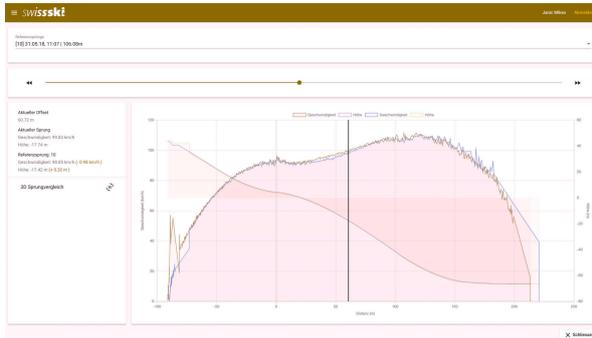
### Material Design Einhaltung

Material Design beinhaltet sehr viele Guides. Angular Material [8], das von Google entwickelt wird, verweist in ihren Designspezifikationen auf die offiziellen Richtlinien [58]. Es wird angenommen, dass die Richtlinien von der Library eingehalten werden.

### Plattformtest

Die Funktionalität der Applikation mit Google Chrome 67 wurde bereits im UAT getestet. Auf dem iPad Pro mit Safari wurden folgende Funktionen getestet:

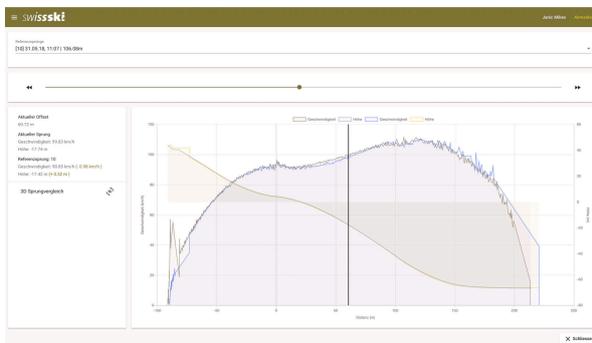
- Sprung Dashboard - Funktioniert. Darstellung fehlerhaft (siehe Abbildung 8.2).
- Messreihe zuweisen - Funktioniert.
- Sprungvergleich 2D - Funktioniert. Inhalt zu breit, Scrollen erforderlich.
- Sprungvergleich 3D - Funktioniert. Restliche Inhalte beim Wechsel auf diesen Modus fehlerhaft (siehe Abbildung 8.4).
- Video Upload - Funktioniert.
- Neuer Sprung hinzufügen - Funktioniert.
- Trainingsübersicht - Funktioniert. Darstellung fehlerhaft (siehe Abbildung 8.3).



(a) Deuteranopia - Grün Blindheit



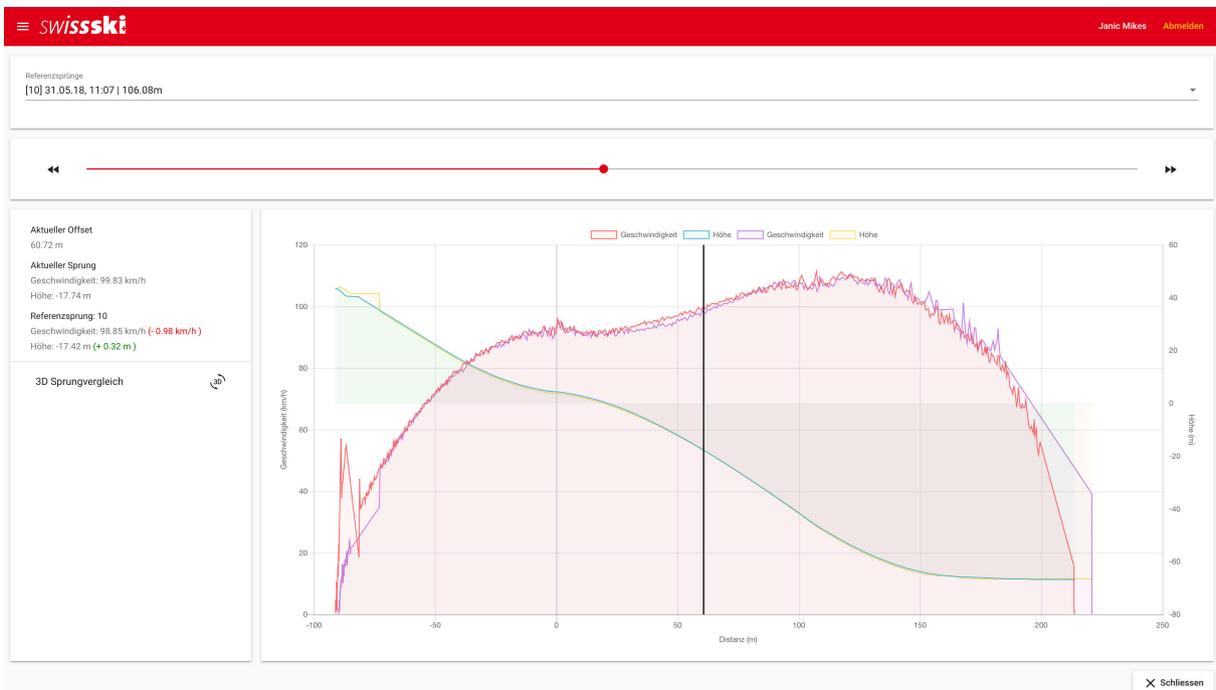
(b) Deuteranomaly - Grün Schwäche



(c) Protanopia - Rot Blindheit



(d) Protanomaly - Rot Schwäche



(e) Original Bild

Abbildung 8.1: Simulierte Farbenblindheit

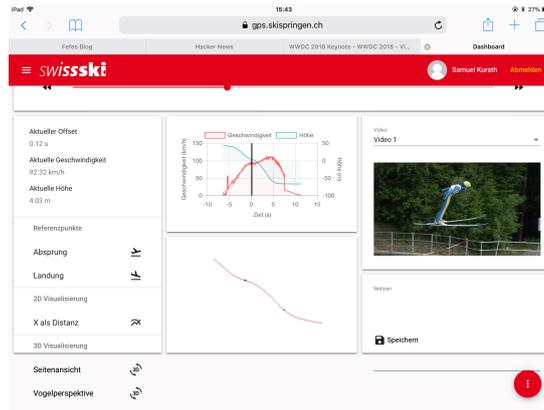


Abbildung 8.2: iPad Screenshot Dashboard

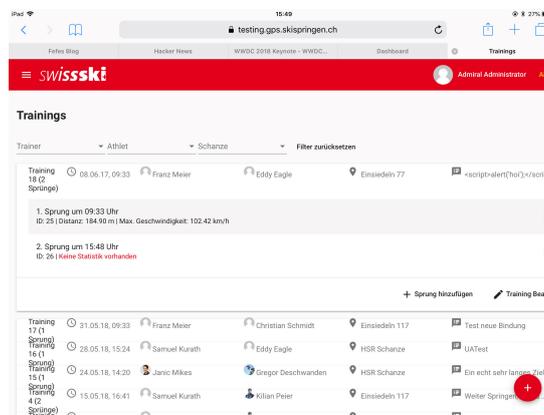


Abbildung 8.3: iPad Screenshot Trainings

### Angular Guidelines

NFA: «Die Coding Guidelines von angular.io werden eingehalten und mit Reviews der Applikation geprüft». Die Angular Guidelines [3] schlagen vor, das Package *codelyzer* zum Linten zu verwenden. Dieser Aufruf wird in Listing 8.1 gezeigt.

```

1 > ng lint
2
3
4 All files pass linting
  
```

Listing 8.1: Angular Linting

Im Code Review (siehe Abschnitt 5.2.1) wurden keine Mängel bezüglich Guidelines festgestellt. Dieser Punkt ist daher erfüllt.

### Django Guidelines

NFA: «Die Coding Guidelines von Django werden eingehalten und mit Reviews der Applikation geprüft».

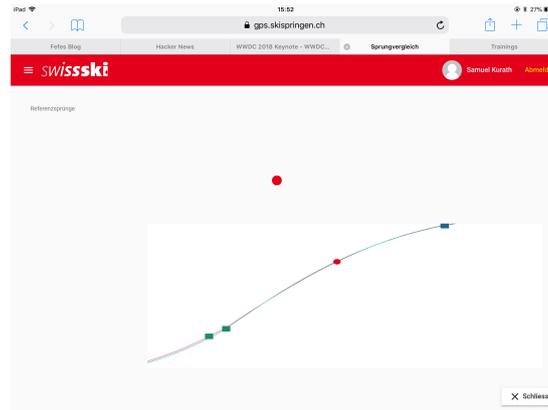


Abbildung 8.4: iPad Screenshot 3D Sprungvergleich

Die Django Guidelines [59] bestehen aus eigenen Vorgaben und wo nichts anderes spezifiziert ist, soll Pep8 eingehalten werden. Daher testen wir den Output von Pep8 Stylechecker pycodestyle in Listing 8.2.

```
1 > pycodestyle  
2 # leerer Output, exit code 0
```

Listing 8.2: Django Linting

Im Review (siehe Abschnitt 5.3.1) wurden keine Mängel bezüglich Guidelines festgestellt. Dieser Punkt ist daher erfüllt.

### Angular Testing

NFA: «Die wichtigsten Komponenten werden mit Unit Tests abgedeckt». Die wichtigsten Komponenten sind Services und die Jump-Detail Komponente. Diese sind mit Unit Tests abgedeckt. Die gesamte Code Coverage beträgt 70%.

NFA: «Im Frontend werden ca. drei End to End (Integration Tests) mit Protractor implementiert». Diese Tests wurden nicht umgesetzt. Ein manueller Integration-Test wurde vor dem UAT mit dem darin beschriebenen Szenario durchgeführt.

### Django Testing

NFA: «Die wichtigsten Komponenten werden mit Unit Tests abgedeckt». Die wichtigsten Komponenten sind die Serializers und die Services. Diese sind mit Unit Tests abgedeckt. Die gesamte Code Coverage beträgt 92% (siehe Listing 5.2).

### Code Review Merge Request

NFA: «Reviews werden für Merge Requests auf Verlangen durchgeführt und protokolliert». Bei Merge Requests wurde über GitLab direkt im Code kommentiert. Solange noch offene Diskussionen bestanden, konnte Request nicht akzeptiert werden. Die Kommentare befinden sich in GitLab Repositories.

## Test Berechtigungen

NFA: «Berechtigungen können an Athleten gegeben und entzogen werden». Athleten haben nur Zugriff auf ihre eigenen Sprünge. Ausserdem können sie diese mit Referenzsprüngen vergleichen. Berechtigungen können nicht auf einzelne Sprünge vergeben werden. Dieser Punkt ist teilweise erfüllt.

NFA: «Sprung kann nur mit entsprechender Berechtigung angezeigt werden». Athleten sehen nur die eigenen Sprünge. Andere Usertypen sehen alle Sprünge. Dieser Punkt ist damit erfüllt.

## Fehlertest

NFA: «Bei auftretenden Fehler werden dem Benutzer diese in verständlicher Form dargestellt».

Test 1: Ungültige URL aufrufen im Frontend. Es erscheint die Meldung *Page not found!*.

Test 2: Ungültige Login Credentials verwenden. Es erscheint die Meldung *Anmeldung fehlgeschlagen*.

Test 3: Es wird das Dashboard eines nicht existierenden Sprunges aufgerufen. Der Ladespinner hört nicht auf zu drehen.

Diese NFA wurde damit teilweise erfüllt.

## 8.2 Use Cases

### Trainer Verwalten (CRUD)

#### Aktoren

Administrator

### Athleten Verwalten (CRUD)

#### Aktoren

Administrator

### Videos Verwalten (CRUD)

#### Aktoren

Administrator

## **Sprünge Verwalten (CRUD)**

### **Aktoren**

Administrator

## **Schanzen Verwalten (CRUD)**

### **Aktoren**

Administrator

## **Zusätzliche Sprung Daten verwalten**

### **Aktoren**

Trainer

### **Voraussetzungen**

- Der Sprung muss bereits erfasst sein

### **Erfolgreiches Szenario**

Beim Ansehen eines Sprunges können weitere Informationen dazu erfasst werden.

## **Videos zu den Sprüngen hinzufügen**

### **Voraussetzungen**

- Der Sprung muss erfasst sein. Das Video ist noch keinem Sprung zugeordnet

### **Erfolgreiches Szenario**

Beim Ansehen eines Sprunges kann der Trainer aus einer Liste auswählen und diese dem Sprung zuweisen.

Post Conditions

Das Video ist dem Sprung zugeordnet.

### **Erweiterungen**

Anstelle der Auswahlliste kann ein Video direkt hochgeladen werden. Optional: Beim Hinzufügen von einem Video können auch bereits zugewiesene Videos angezeigt werden. Diese werden dann vom alten Sprung entfernt und neu zugewiesen.

## Videos von Sprung entfernen

### Aktoren

Trainer

### Voraussetzungen

- Das Video ist einem Sprung zugewiesen

### Erfolgreiches Szenario

Beim Ansehen eines Sprunges sieht man die zugeordneten Videos. Dabei kann die Verbindung von einem Video entfernt werden.

## Video Referenzpunkt festlegen

### Aktoren

Trainer

### Voraussetzungen

- Das Video muss erfasst sein.

### Erfolgreiches Szenario

Beim Anzeigen des Videos kann dessen Referenzzeitpunkt (Schanzentisch, Absprung) festgelegt werden. Diese Zeit wird mit dem Video gespeichert.

## Sprung Referenzpunkt festlegen

### Aktoren

Trainer

### Voraussetzungen

- Der Sprung muss erfasst sein.

### Erfolgreiches Szenario

Beim Anzeigen des Sprunges kann dessen Referenzmesspunkt (Schanzentisch, Absprung) festgelegt werden. Dieser Punkt wird mit dem Sprung gespeichert.

## Videos von Sprüngen mit Sprungdaten synchronisiert haben

### Aktoren

Trainer, Athlet

### Voraussetzungen

- Das Video hat einen Referenzpunkt zugewiesen.
- Der Sprung hat einen Referenzpunkt zugewiesen

### Erfolgreiches Szenario

Beim Ansehen des Videos wird bei den Sprungdaten der jeweilige Punkt hervorgehoben, welcher den Zeitpunkt repräsentiert.

## Sprünge einem Athleten zuweisen

### Aktoren

Trainer

### Voraussetzungen

- Der Athlet muss erfasst sein.
- Der Sprung muss vom erfasst sein.

### Erfolgreiches Szenario

Beim Erfassen eines Sprunges wählt der Aktor einen Athleten aus, um diesem den Sprung zuzuweisen.

## Sichtbarkeit von Sprüngen für die Athleten zuweisen

### Aktoren

Trainer

### Voraussetzungen

- Der Sprung muss erfasst sein.
- Der Athlet muss erfasst sein.

### **Erfolgreiches Szenario**

Bei der Ansicht eines Sprungs kann der Aktor weiteren Athleten, Zugriffsrechte gewähren.

### **Synchronisierung von Video und Sprung anhand von automatisch erkannten Wendepunkten wählen**

Optional

#### **Aktoren**

Trainer

#### **Voraussetzungen**

- Der Sprung muss erfasst sein.

### **Erfolgreiches Szenario**

Beim Festlegen des Referenzpunktes eines Sprunges kann der Aktor von Punkten auswählen, welche automatisch vorgeschlagen werden.

### **Einen Sprung als Muster, Referenz festlegen**

Optional

#### **Aktoren**

Trainer

#### **Voraussetzungen**

- Der Sprung muss erfasst sein.

### **Erfolgreiches Szenario**

Ein geöffneter Sprung kann als Referenzsprung gekennzeichnet werden. Sprungposition zwischen Trainer und Athlet synchronisieren Optional

#### **Aktoren**

Trainer, Athlet

## **Erfolgreiches Szenario**

Als Trainer kann ich eine Sync Session starten, welche vom Athleten geöffnet werden kann. Sobald der Trainer die Zeit im Video oder Sprung anpasst, wird beim Athleten dieselbe Zeit angezeigt.

## **Einen Sprung ansehen**

### **Aktoren**

Trainer, Athlet

### **Voraussetzungen**

- Der Sprung muss erfasst sein.

## **Erfolgreiches Szenario**

Aus einer Liste von Sprüngen kann ein einzelner ausgewählt werden. Die Sprungdaten des ausgewählten Sprungs werden angezeigt. Der Aktor kann aus verschiedenen voreingestellten Kameraperspektiven die Ansicht des 3D Modells beeinflussen. Das 2D Modell kann mittels der Maus beeinflusst werden. Das 2D- und 3D Modell sowie das Video sollen miteinander synchronisiert werden.

## **Mehrere Sprünge vergleichen können**

### **Aktoren**

Trainer, Athlet

### **Voraussetzungen**

- Alle Sprünge welche verglichen werden möchten, müssen erfasst sein.

## **Erfolgreiches Szenario**

Beim Ansehen von einem Sprung können weitere Sprünge als Vergleich hinzugefügt werden. Die Flugbahnen werden in verschiedenen Farben gleichzeitig dargestellt.

## **Videos verlangsamt anschauen**

### **Aktoren**

Trainer, Athlet

## Voraussetzungen

- Das Video muss erfasst sein.

## Erfolgreiches Szenario

Beim Ansehen eines Videos kann der Aktor mit Schaltflächen schrittweise durch ein Video steppen. Extended Optional: Beim Ansehen eines Videos kann der Aktor mit Schaltflächen die Wiedergabegeschwindigkeit des Videos einstellen.

## Geschwindigkeit gleichzeitig mit dem Video sehen

### Aktoren

Trainer, Athlet

### Voraussetzungen

- Das Video muss erfasst sein.
- Der Sprung muss erfasst sein.

## Erfolgreiches Szenario

Im Trajectory Dashboard kann der Aktor das Video und die aktuelle Geschwindigkeit gleichzeitig sehen.

## Die Sprünge gruppiert nach Trainings / Athlet sehen

### Aktoren

Trainer, Athlet

### Voraussetzungen

- Die Sprünge müssen erfasst sein.
- Die Trainings müssen erfasst sein.
- Die Athleten müssen erfasst sein.

## Erfolgreiches Szenario

In der Trajectory-Overview kann der Aktor die Sprünge nach Training und/oder Athlet filtern.

## 8.3 UI Mockups

Nachfolgend sind 14 UI Mockups zu finden.

02-dashboard

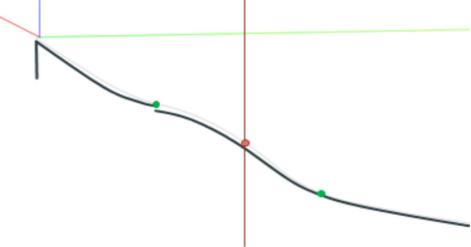
Interaktive Skisprung-Trainingsanalyse - Neuer Sprung

← → ↻

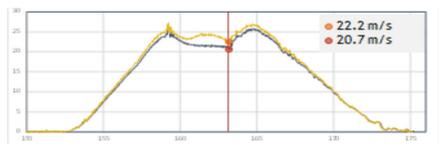
≡ **Swisski** Neues Training 

### Dashboard Sprung vom 13.04.2018 - Peter Klingler - GS Einsiedeln

Video 2 Seite Vogel Frontal

Referenzsprung 1



Distanz	210m
Absprungsgeschwindigkeit	66 km/h
Landegeschwindigkeit	72 km/h
Höchstgeschwindigkeit	76 km/h
Wettereinfluss	3/5
Sprungbewertung	4/5

Fabelhafter Sprung von Peter.  
Alles passte. Der Wind war sehr vorteilhaft.



03-popover-new-training

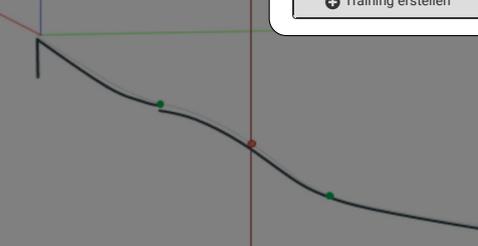
Interaktive Skisprung-Trainingsanalyse - Neuer Sprung

← → ↻

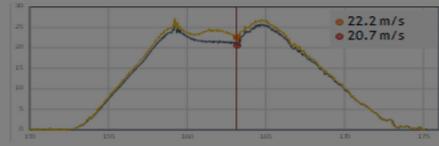
≡ **Swisski** Neues Training 

### Dashboard Sprung vom 13.04.2018 - Peter Klingler - GS Einsiedeln

Video 2 Seite Vogel

Referenzsprung 1



Distanz	210m
Absprungsgeschwindigkeit	66 km/h
Landegeschwindigkeit	72 km/h
Höchstgeschwindigkeit	76 km/h
Wettereinfluss	3/5
Sprungbewertung	4/5

Guter Sprung bei schwierigen Windverhältnissen.  
Dennoch wurde eine gute Weite erreicht.

[3.1s] Sehr guter absprung, idealer Zeitpunkt.  
[4.7s] Die Arme näher am Körper halten, Hände Flach!



Athlet auswählen

Schanze auswählen

 Training erstellen

04-neuer-sprung

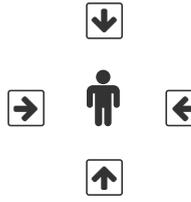
Interaktive Skisprung-Trainingsanalyse - Neuer Sprung

← → ↻

☰ **Swisski** Neues Training + ⋮

### Neuer Sprung

Windeinfluss



Sprungbewertung

★★★★☆

Gate

### Notizen

Guter Sprung bei schwierigen Windverhältnissen.  
Dennoch wurde eine gute Weite erreicht.

05-dashboard-neuer-sprung

Interaktive Skisprung-Trainingsanalyse - Neuer Sprung

← → ↻

☰ **Swisski** Neues Training + ⋮

### Dashboard Sprung vom 12.04.2018 - Alessandro Schmid - KS Einsiedeln

kein Video erfasst





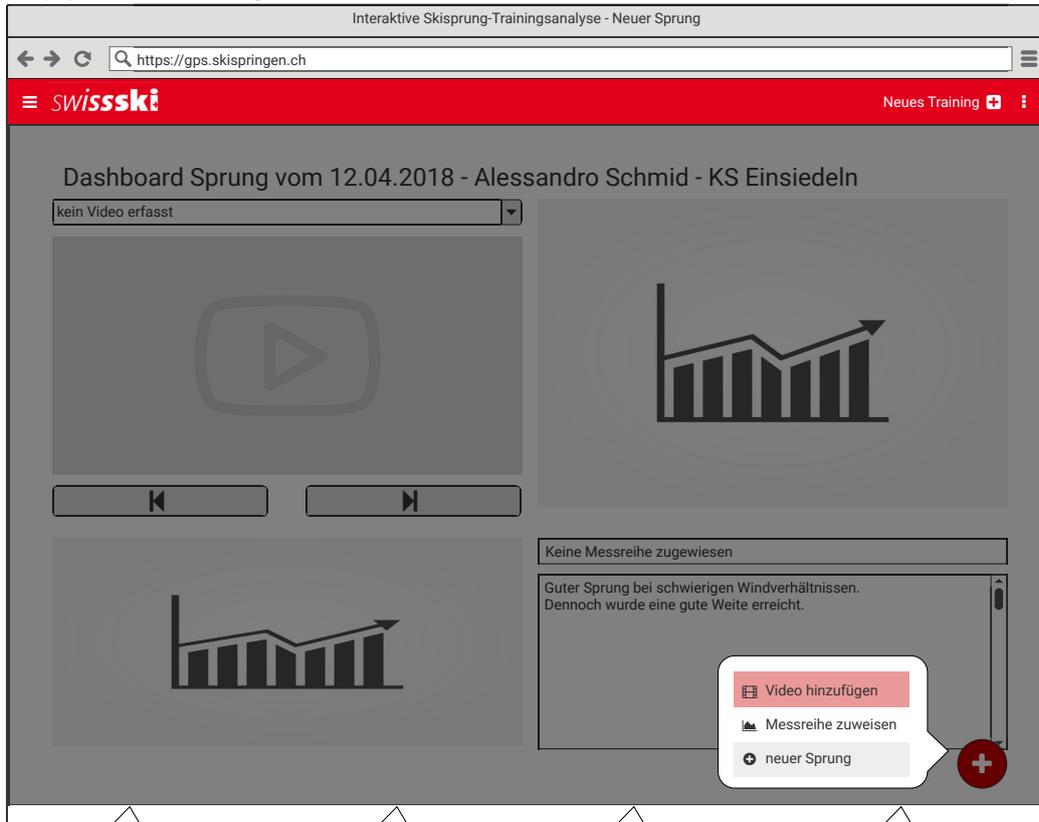


Keine Messreihe zugewiesen

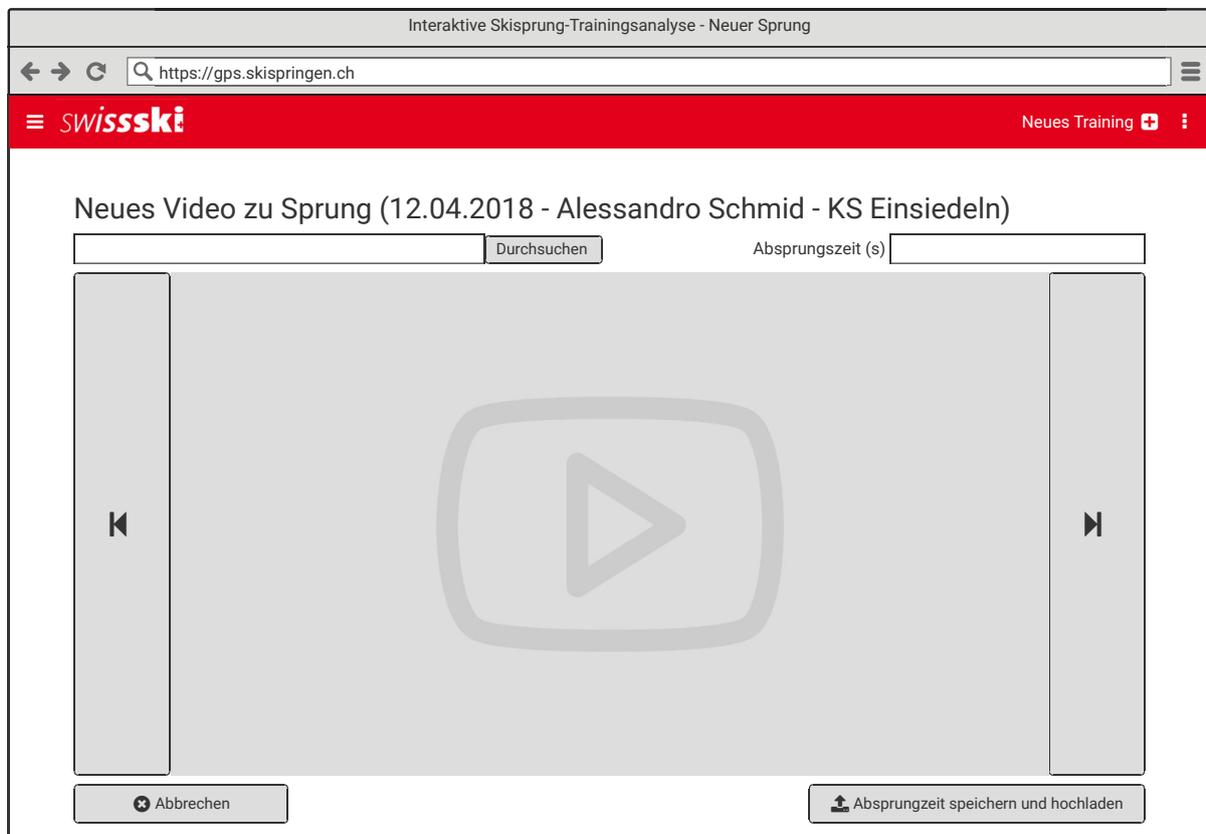
Guter Sprung bei schwierigen Windverhältnissen.  
Dennoch wurde eine gute Weite erreicht.



06-popover-etwas-hinzufügen



07-video-hinzufügen



07-b-video-hinzufügen

Interaktive Skisprung-Trainingsanalyse - Neuer Sprung

https://gps.skispringen.ch

swisski Neues Training

### Neues Video zu Sprung (12.04.2018 - Alessandro Schmid - KS Einsiedeln)

video\_1124.mp4  Absprungszeit (s) 3.1025



08-dashboard-mit-video

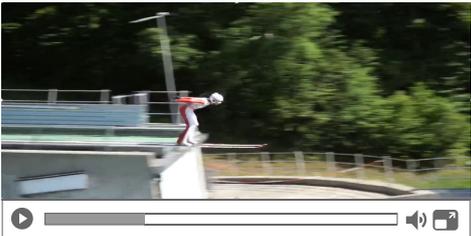
Interaktive Skisprung-Trainingsanalyse - Neuer Sprung

https://gps.skispringen.ch

swisski Neues Training

### Dashboard Sprung vom 12.04.2018 - Alessandro Schmid - KS Einsiedeln

Video 1



Keine Messreihe zugewiesen

Guter Sprung bei schwierigen Windverhältnissen.  
Dennoch wurde eine gute Weite erreicht.



10-sprung-hinzufügen

Interaktive Skisprung-Trainingsanalyse - Neuer Sprung

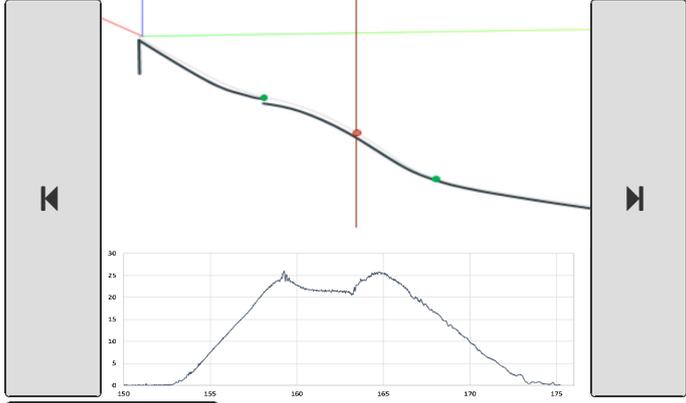
https://gps.skispringen.ch


Neues Training 

### Messreihe zuordnen für Sprung (12.04.2018 - Alessandro Schmid - KS Einsiedeln)

Datum	Uhrzeit	Sprung-ID
12.04.2018	10:34	[4]
10.04.2018	14:42	[2]
10.04.2018	13:37	[1]

Start
Absprung
Landung



11-sprung-dashboard

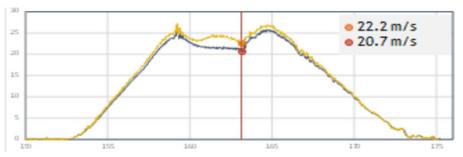
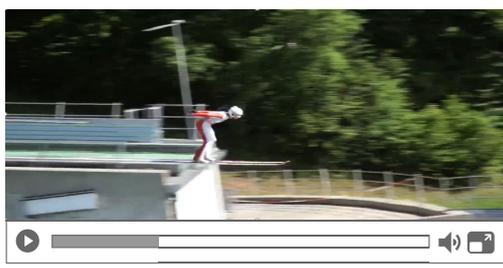
Interaktive Skisprung-Trainingsanalyse - Neuer Sprung

https://gps.skispringen.ch

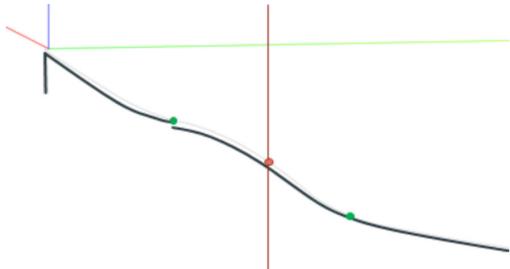

Neues Training 

### Dashboard Sprung vom 22.04.2018 - Alessandro Schmid - KS Einsiedeln

Video 1



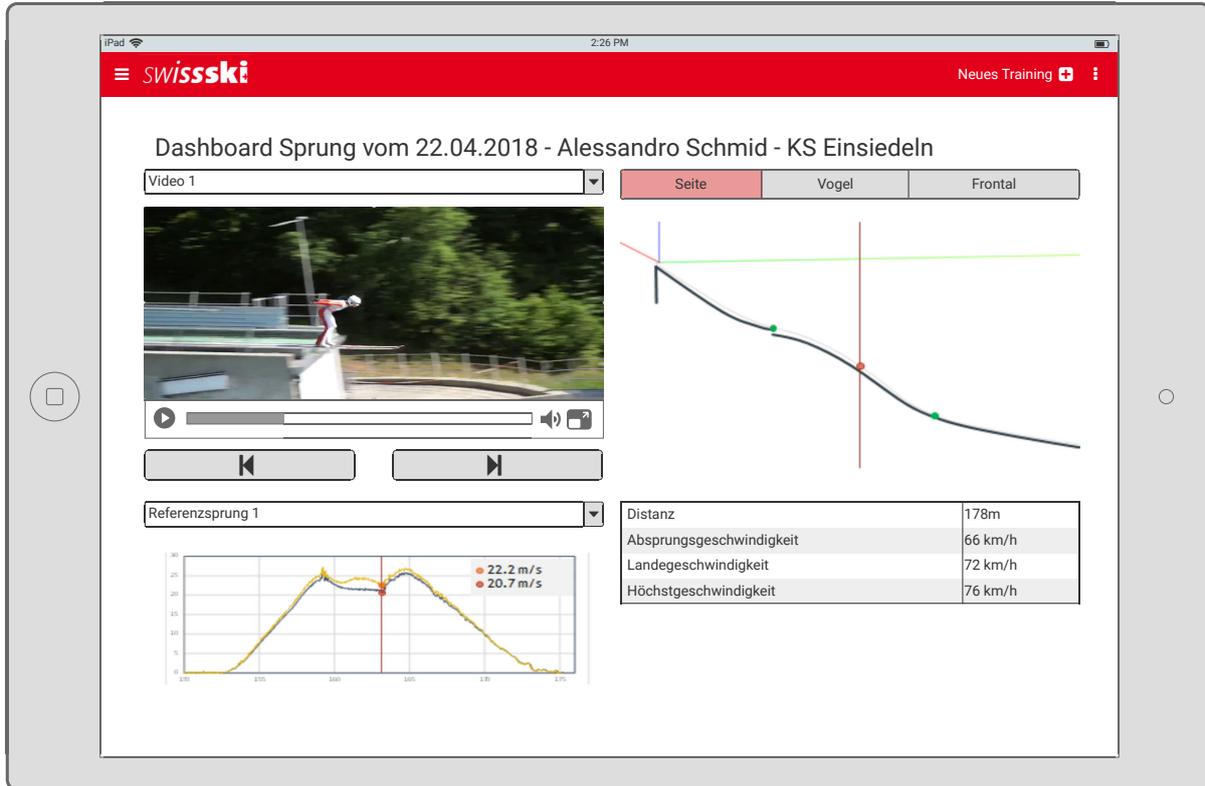
Seite
Vogel
Frontal



Distanz	178m
Absprungsgeschwindigkeit	66 km/h
Landegeschwindigkeit	72 km/h
Höchstgeschwindigkeit	76 km/h
Wettereinfluss	3/5
Sprungbewertung	4/5

Guter Sprung bei schwierigen Windverhältnissen. Dennoch wurde eine gute Weite erreicht.

13-sprung-dashboard-athlete



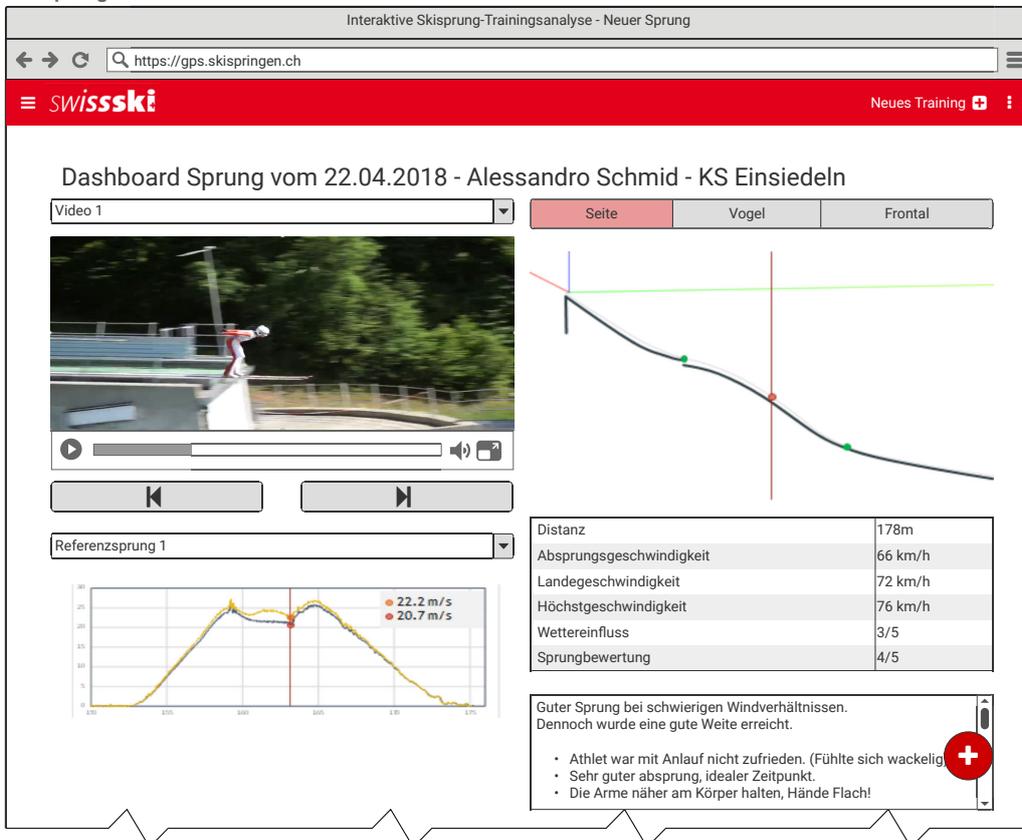
Dashboard Sprung vom 22.04.2018 - Alessandro Schmid - KS Einsiedeln

Video 1 | Seite | Vogel | Frontal

Referenzsprung 1

Distanz	178m
Absprunggeschwindigkeit	66 km/h
Landegeschwindigkeit	72 km/h
Höchstgeschwindigkeit	76 km/h

14-sprung-dashboard-fin



Interaktive Skisprung-Trainingsanalyse - Neuer Sprung

Dashboard Sprung vom 22.04.2018 - Alessandro Schmid - KS Einsiedeln

Video 1 | Seite | Vogel | Frontal

Referenzsprung 1

Distanz	178m
Absprunggeschwindigkeit	66 km/h
Landegeschwindigkeit	72 km/h
Höchstgeschwindigkeit	76 km/h
Wettereinfluss	3/5
Sprungbewertung	4/5

Guter Sprung bei schwierigen Windverhältnissen. Dennoch wurde eine gute Weite erreicht.

- Athlet war mit Anlauf nicht zufrieden. (Fühlte sich wackelig)
- Sehr guter absprung, idealer Zeitpunkt.
- Die Arme näher am Körper halten, Hände Flach!

15-sprung-bearbeiten

Interaktive Skisprung-Trainingsanalyse - Neuer Sprung

← → ↻

≡ swisski Neues Training

### Sprung xyz bearbeiten

Windeinfluss  Trajectory 01

★ ★ ★ ☆ ☆

Sprungbewertung

★ ★ ★ ★ ☆

Gate

Notizen

Guter Sprung bei schwierigen Windverhältnissen.  
Dennoch wurde eine gute Weite erreicht.

Fertigstellen

Sprünge

Interaktive Skisprung-Trainingsanalyse

← → ↻

≡ swisski Neues Training

Nur meine Trainings

Item One

Item Two

Item Three

## 8.4 Continuous Integration und Deployment

### 8.4.1 GitLab Runner

GitLab Runner wurde im BA-Zimmer 1.262 auf dem freien Computer pin1262018 installiert. Als Betriebssystem wurde Debian 9.4 *stretch* verwendet. Der Runner wurde gemäss Anleitung [60] installiert. Um diese Features weiter nutzen zu können, muss entweder auf die shared Runners von GitLab umgestiegen, oder ein eigener Runner aufgesetzt werden, welcher sich um die Jobs kümmert. Dieser könnte auch direkt auf dem Server von Swiss-Ski aufgesetzt werden.

```
1 root@pin1262018:~# cat /etc/gitlab-runner/config.toml
2 concurrent = 10
3 check_interval = 0
4
5 [[runners]]
6   name = "pin1262018"
7   url = "https://gitlab.com/"
8   token = "xxx"
9   executor = "docker"
10  [runners.docker]
11    tls_verify = false
12    image = "alpine"
13    privileged = true
14    disable_cache = false
15    volumes = ["/cache"]
16    shm_size = 0
17  [runners.cache]
```

Listing 8.3: GitLab Runner Konfiguration

Die Einstellung *privileged* muss auf *true* gesetzt werden, da der Build der Base Images selbst auch Docker benötigt.

### 8.4.2 Eigene Base Images

Um die CI Tasks zu beschleunigen, werden eigene Base Images verwendet, die die notwendigen Systemlibraries bereits beinhalten. Das Listing 8.4 beinhaltet das Dockerfile, welches für den Angular Client Build verwendet wird.

```
1 FROM node:9
2 RUN wget -q -O - https://dl-ssl.google.com/linux/linux_signing_key.pub | apt-key
   ↪ add - && \
3   echo "deb http://dl.google.com/linux/chrome/deb/ stable main" >> /etc/apt/
   ↪ sources.list && \
4   apt-get update && \
5   apt-get install --no-install-recommends -y \
6     google-chrome-stable \
7     gconf-service \
8     libasound2 \
```

```
9      libatk1.0-0 \  
10     libc6 \  
11     libcairo2 \  
12     libcups2 \  
13     libdbus-1-3 \  
14     libexpat1 \  
15     libfontconfig1 \  
16     libgcc1 \  
17     libgconf-2-4 \  
18     libgdk-pixbuf2.0-0 \  
19     libglib2.0-0 \  
20     libgtk-3-0 \  
21     libnspr4 \  
22     libpango-1.0-0 \  
23     libpangocairo-1.0-0 \  
24     libstdc++6 \  
25     libx11-6 \  
26     libx11-xcb1 \  
27     libxcb1 \  
28     libxcomposite1 \  
29     libxcursor1 \  
30     libxdamage1 \  
31     libxext6 \  
32     libxfixed3 \  
33     libxi6 \  
34     libxrandr2 \  
35     libxrender1 \  
36     libxss1 \  
37     libxtst6 \  
38     ca-certificates \  
39     fonts-liberation \  
40     libappindicator1 \  
41     libnss3 \  
42     lsb-release \  
43     xdg-utils \  
44     wget && \  
45     rm -rf /var/lib/apt/lists/* \  
46     && apt-get purge --auto-remove -y curl \  
47     && rm -rf /src/*.deb  
48  
49 USER node:node
```

Listing 8.4: Frontend CI Dockerfile

In Listing 8.5 ist der CI Task aufgeführt, der das Angular Base Image in die GitLab Registry hochlädt. Dieser Task wird nicht nur bei einem neuen Commit auf das Repository ausgeführt, sondern auch täglich.

```
1 node_build_image_ci:  
2   image: docker:git  
3   services:  
4   - docker:dind  
5   script:
```

```
6 - docker login -u gitlab-ci-token -p $CI_BUILD_TOKEN registry.gitlab.com
7 - docker build -t registry.gitlab.com/ba-skijump/ski-baseimages/node/ci:latest
8 ↪ -f ./node/Dockerfile.ci ./node
9 - docker push registry.gitlab.com/ba-skijump/ski-baseimages/node/ci:latest
10 only:
    - master
```

Listing 8.5: Autobuild Build Images

### 8.4.3 Buildfiles

In den zu buildenden Repositories wurde jeweils ein *.gitlab-ci.yml* File erstellt. In Listing 8.6 befindet sich ein Auszug aus dem Client Repository.

```
1 image: registry.gitlab.com/ba-skijump/ski-baseimages/node/ci:latest
2 stages:
3   - build
4   - test
5   - deploy
6 build:
7   stage: build
8   only:
9     - production
10 cache:
11   paths:
12     - node_modules/
13 script:
14   - yarn --frozen-lockfile
15   - yarn run build --env=prod
16 artifacts:
17   paths:
18     - dist/
19 production:
20   stage: deploy
21   environment:
22     name: Production
23     url: https://gps.skispringen.ch/
24   when: manual
25   only:
26     - production
27   script:
28     - .scripts/setup_ssh.sh
29     - touch dist/.static
30     - tar c dist | ssh dokku@gps.skispringen.ch tar:in production
```

Listing 8.6: CI File

## 8.5 Server Umgebung

Die Lösung wurde auf dem Cloudscale.ch VPS `gps.skispringen.ch` installiert. Dieser läuft auf Ubuntu 16.04.4 LTS.

### 8.5.1 Dokku

Die PaaS Lösung Dokku wurde in der Version 0.11.6 durch Samuel Kurath installiert. [61] Zusätzlich wurden die Plugin dokku-postgres [62] sowie dokku-letsencrypt [57] installiert.

Für Dokku wurde im DNS folgender Record erstellt:

```
1 .gps.skispringen.ch IN CNAME gps.skispringen.ch.
```

Listing 8.7: DNS Eintrag

Damit werden alle Subdomains von `gps.skispringen.ch` auf den Server geleitet.

### 8.5.2 Aufsetzen Backend App

Für jede Instanz wurde folgende Prozedur durchgeführt.

#### Applikation

```
1 dokku apps:create api
2 ----> Creating api... done
3 dokku config:set --no-restart api BUILD_WITH_GEO_LIBRARIES='1'
```

Listing 8.8: App erstellen

Nun kann die Applikation über Git deployed werden. Dazu muss Dokku als Remote Repository im Git eingetragen und dann gepusht werden.

```
1 git remote add dokku dokku@gps.skispringen.ch:api
2 git push dokku master
```

Listing 8.9: App erstellen

Nach einem erfolgreichen ersten Deployment kann Let's encrypt aktiviert werden.

```
1 dokku config:set --no-restart api DOKKU_LETSENCRYPT_EMAIL=xxx@hsr.ch
2 dokku letsencrypt api
```

Listing 8.10: letsencrypt aktivieren

#### Datenbank

```
1 docker pull mdillon/postgis:10
2 export POSTGRES_IMAGE="mdillon/postgis"
3 export POSTGRES_IMAGE_VERSION="10"
4
5 dokku postgres:create api-db
6     Waiting for container to be ready
7     Creating container database
8     Securing connection to database
9     =====> Postgres container created: api-db
10    =====> Container Information
11    Config dir:           /var/lib/dokku/services/postgres/api-db/config
12    Data dir:            /var/lib/dokku/services/postgres/api-db/data
13    Dsn:                 postgres://postgres:xxx@dokku-postgres-api-db:5432/api_db
14    Exposed ports:       -
15    Id:                  xxx
16    Internal ip:         172.17.0.xxx
17    Links:               -
18    Service root:       /var/lib/dokku/services/postgres/api-db
19    Status:              running
20    Version:             mdillon/postgis:10
21
22 dokku postgres:link api-db api
23     -----> Setting config vars
24     DATABASE_URL: postgres://postgres:xxx@dokku-postgres-api-db:5432/api_db
25     -----> Restarting app test-api
```

Listing 8.11: Datenbank aufsetzen

Nun muss die Datenbank URL angepasst werden, damit Django PostGIS verwendet. Dazu wird der *postgres* Prefix durch *postgis* ersetzt.

```
1 dokku config:set api DATABASE_URL=postgis://postgres:xxx@dokku-postgres-api-db
   ↪ :5432/api_db
```

Listing 8.12: Datenbank URL Anpassen

Soll das Backend im Debug Modus laufen, muss auch die Environment Variable DEBUG auf 1 gesetzt werden.

### Nginx Anpassungen

Die Videos müssen direkt von Nginx bereitgestellt werden, da der verwendete Python Webserver keine HTTP Range Requests unterstützt. Ausserdem soll die maximale Uploadgrösse von 1 MB erhöht werden, um grössere Videofiles zu unterstützen.

```
1 ubuntu@GPS-Skispringen:/home/dokku/api/nginx.conf.d$ cat mediafiles.conf
2 location /media/ {
3     alias /mnt/bulk/dokku/api-prod/mediafiles/;
4 }
5
6 ubuntu@GPS-Skispringen:/home/dokku/api/nginx.conf.d$ cat upload.conf
7 client_max_body_size 100m;
```

```
8  
9 systemctl reload nginx.service
```

Listing 8.13: Nginx Anpassungen

## Persistent Storage

Damit die hochgeladenen Dateien auch zwischen zwei Deploys erhalten bleiben, muss in Dokku ein Mount konfiguriert werden.

```
1 mkdir /mnt/bulk/dokku/api-prod/  
2 chown dokku:dokku /mnt/bulk/dokku/api-prod  
3 mkdir /mnt/bulk/dokku/api-prod/mediafiles  
4 chown 32767:32767 /mnt/bulk/dokku/api-prod/mediafiles  
5  
6 dokku storage:mount api /mnt/bulk/dokku/api-prod/mediafiles/:/app/mediafiles  
7 dokku storage:report api  
8 =====> api storage information  
9     Storage build mounts:  
10     Storage deploy mounts: -v /mnt/bulk/dokku/api-prod/mediafiles/:/app/  
    ↪ mediafiles  
11     Storage run mounts: -v /mnt/bulk/dokku/api-prod/mediafiles/:/app/  
    ↪ mediafiles
```

Listing 8.14: Dokku Storage für Mediafiles

## 8.5.3 Aufsetzen Client App

Die Domain wird für die Produktion manuell gesetzt, da sie direkt unter *gps.skispringen.ch* sein soll.

### Applikation

```
1 dokku apps:create production  
2 dokku domains:set production gps.skispringen.ch
```

Listing 8.15: Dokku Angular App erstellen

Die Client Applikation wird über tar deployed, da der Buildprozess unserer Meinung nach etwas langsam ist und wir dank CI bereits eine kompilierte Version zur Verfügung haben.

```
1 yarn install --frozen-lockfile  
2 yarn run build --env=prod  
3 touch dist/.static  
4 tar c dist | ssh dokku@gps.skispringen.ch tar:in production
```

Listing 8.16: Client deployen

Nach dem ersten Deployment kann auch im Client Let's encrypt aktiviert werden.

```
1 dokku config:set --no-restart production DOKKU_LETSENCRYPT_EMAIL=xxx@hsr.ch
2 dokku letsencrypt production
```

Listing 8.17: Let's encrypt aktivieren

## 8.5.4 Backup Restore

Nur die Backend Applikation beinhaltet Nutzdaten, die gesichert werden können.

### Datenbank

Ein Backup der Datenbank *api-db* wird mit dem Befehl *dokku postgres:export api-db > api.sql* in das File *api.sql* durchgeführt. Mit *dokku postgres:import api-db > api.sql* wird das Backup wieder eingespielt. Mit *dokku postgres:list* erhält man eine Liste aller vorhandenen Datenbanken.

### Mediafiles

Die Mediafiles befinden sich im Ordner */mnt/bulk/dokku/*. Dieser Ordner kann mit normalen Shellkommandos gesichert werden.

### Dokku

Falls die Dokku Applikationen nicht mehr vorhanden sind, können Sie wie in Abschnitt 8.5 wieder neu aufgesetzt werden. Danach werden die obigen Schritte durchgeführt.

## 8.5.5 Filewatcher

Um neue Messdaten automatisch in die produktive Umgebung einzuspielen, muss der Filewatcher auf dem Server installiert werden. Er befindet sich im Backendrepository im Ordner *scripts*.

### Abhängigkeiten

Filewatcher benötigt die Python Libraries *inotify* und *requests*.

### Installation

Wir kopierten dieses File nach */opt/filewatcher/watcher.py*. In Listing 8.18 ist das Systemd Unitfile, welches wir in */etc/systemd/system/filewatcher.service* gespeichert haben.

```
1 [Unit]
2 Description=Posfile Watcher for api.gps.skispringen.ch
3 After=syslog.target
4
5 [Service]
```

```
6 Type=simple
7 User=syncboy
8 Group=syncboy
9 WorkingDirectory=/opt/filewatcher
10 ExecStart=/opt/filewatcher/watcher.py -b 'https://api.gps.skispringen.ch' -d /home
    ↪ /syncboy/GNSS/Solution -u filewatcher -p 'xxx'
11 StandardOutput=syslog
12 StandardError=syslog
13
14 [Install]
15 WantedBy=multi-user.target
```

Listing 8.18: Filewatcher Systemd Unitfile

## Service

Mit `systemctl` Befehl `filewatcher.service` kann der Dienst gesteuert werden. Die wichtigsten Befehle sind `status`, `start` und `stop`.

## 8.6 Entwicklungsumgebung

Der Code ist auf GitLab gehostet. Die Aufteilung der Repositories ist in Tabelle 8.2 aufgeführt.

Name	Zweck	URL
client	Angular Applikation	<a href="https://gitlab.com/ba-skijump/client">https://gitlab.com/ba-skijump/client</a>
backend	Django Applikation	<a href="https://gitlab.com/ba-skijump/backend">https://gitlab.com/ba-skijump/backend</a>
documentation	LaTeX Dokumentation	<a href="https://gitlab.com/ba-skijump/documentation">https://gitlab.com/ba-skijump/documentation</a>
ski-baseimages	Dockerfiles Baseimage	<a href="https://gitlab.com/ba-skijump/ski-baseimages">https://gitlab.com/ba-skijump/ski-baseimages</a>

Tabelle 8.2: Git Repositories

### 8.6.1 Frontend

Das Frontend wurde mit `WebStorm` 2018.1 auf Windows und macOS entwickelt.

#### System Requirements

Um ein Angular Projekt mit der CLI zu erstellen und zu bearbeiten, muss `Node.js` ab Version 8 sowie `Yarn` als `Node.js` Package Manager installiert werden, siehe Angular Quickstart Guide [63].

Wir verwenden für dieses Projekt `Yarn`, da sich herausstellte, dass die Installation der Abhängigkeiten spürbar schneller vonstatten ging.

Um mit Angular CLI zu arbeiten, installieren wir es als globales Package.

```
1 yarn add --global @angular/cli
```

Listing 8.19: Globale Node Module

Nach dem Klonen des Repositories werden die Abhängigkeiten mit dem Befehl `yarn install` installiert.

Um die Applikation lokal laufen zu lassen, führt man `yarn start` aus. Nun wird mit Webpack die App zusammengebaut und mit einem integrierten Webserver auf `localhost:4200` bereitgestellt.

### API anpassen

Ohne Anpassungen geht die App im lokalen Modus auf die Test API-Instanz auf dem Server von Swiss-Ski. Um diese auf eine Lokale API umzuleiten, kann die URL unter `src/environments/environment.ts` angepasst werden.

## 8.6.2 Backend

Das Backend wurde mit `PyCharm` 2018.1 (Professional Edition) auf `Debian` 9.4 *stretch* entwickelt. Für das lokale Entwickeln wird `SQLite` mit der `Spatialite` Extension verwendet. Dadurch wird keine lokale `PostgreSQL` Instanz benötigt. Alternativ kann `PyCharm` auch eine Python Umgebung aus `Docker` verwenden. Für dieses Szenario befindet sich im Repository die Datei `docker-compose.pycharm.yml`. Weitere Details befinden sich im README.

### System Requirements

Folgende Packages müssen installiert werden.

```
1 apt-get install binutils libproj-dev gdal-bin libsqlite3-mod-spatialite spatialite  
↔ -bin
```

Listing 8.20: Debian Packages

Auf Debian muss Python 3.6.4 manuell installiert werden. Die `loadable-sqlite-extensions` werden für `Spatialite` benötigt.

```
1 wget https://www.python.org/ftp/python/3.6.4/Python-3.6.4.tgz  
2 tar xf Python-3.6.4.tgz  
3 cd Python-3.6.4  
4 ./configure --enable-loadable-sqlite-extensions  
5 make  
6 make altinstall
```

Listing 8.21: Manuelle Python Installation

Der Package Manager `Pipenv` kann nun mit dem neuen Python installiert werden.

```
1 pip3.6 install pipenv
```

Listing 8.22: Python Package Manager

### Application Requirements

Um die für die Entwicklung benötigten Libraries zu installieren, sind folgende Befehle im Repository Root auszuführen.

```
1 python3.6 -m venv .venv  
2 pipenv install --dev
```

Listing 8.23: Virtualenv

### IDE Konfiguration

In dem nun vorbereiteten Repository kann PyCharm geöffnet werden. Es soll das Virtualenv automatisch erkennen. Ansonsten kann dies in der Option *Project Interpreter* eingetragen werden. Bei den Run Configurations ist die Environment Variable *DEBUG* auf *1* zu setzen.

### Wichtige Befehle

```
1 # Eine interaktive Shell im Virtual Environment öffnen  
2 pipenv shell  
3 # Datenbankmigrationen durchführen oder initiale Datenbank erstellen  
4 python manage.py migrate  
5 # Django Server lokal laufen lassen  
6 python manage.py runserver
```

Listing 8.24: Wichtige Django Befehle

## 8.7 Benutzeranleitung

Die Applikation läuft auf <https://gps.skispringen.ch>.

Beim ersten Besuch der Web Applikation wird der Login Screen angezeigt. Auf diesem meldet man sich mit den Benutzerdaten an, welche man vom Systemadministrator bekommen hat.

### Verwendung von Begriffen

Ein Training ist eine Trainingseinheit auf genau einer Schanze mit genau einem Athleten und einem Trainer.

Wenn man auf eine andere Schanze wechselt oder ein anderer Athlet messen möchte, so muss ein neues Training erstellt werden.

Ein Sprung muss immer unter einem bestehenden Training erstellt werden.

Die Anzahl an Sprüngen pro Training ist nicht begrenzt. Es wird jedoch empfohlen, ein eigenes Training pro Tag zu erstellen.

Anmerkungen und Notizen können auf dem Training als Trainingsziel oder zu einem Sprung als Sprungnotizen erfasst werden.

Die Interaktion geschieht über den Slider oder über die Pfeiltasten auf der Tastatur. Links und Rechts entsprechen einzel Schritten. Oben und Unten machen grosse Schritte.

### Berechtigungen & Einschränkungen

- Athleten sehen nur ihre eigenen Trainings und Sprünge
- Trainer sehen alle Trainings und Sprünge.
- Um ein Training zu verwalten, müssen Sie als Trainer angemeldet sein.

#### 8.7.1 Trainingsverwaltung

##### Trainingsübersicht

Um in die Trainingsübersicht zu gelangen, öffnet man entweder das Side-Menu über den Menu Button (drei Striche ganz links in der Toolbar), oder man klickt auf das Swiss-Ski Logo.

##### Neues Training erstellen

Ein neues Training kann entweder in der Trainingsübersicht oder vom Sprungdashboard aus über den Action Button (Abbildung 8.6) erstellt werden.

Danach kann der Athlet, die Schanze sowie das Trainingsziel eingetragen werden und mit *Training hinzufügen* gespeichert werden.

Es kann auch ein Training für einen anderen Trainer eröffnet werden.

Nach dem Hinzufügen steht automatisch ein neuer Sprung zum Erfassen bereit.

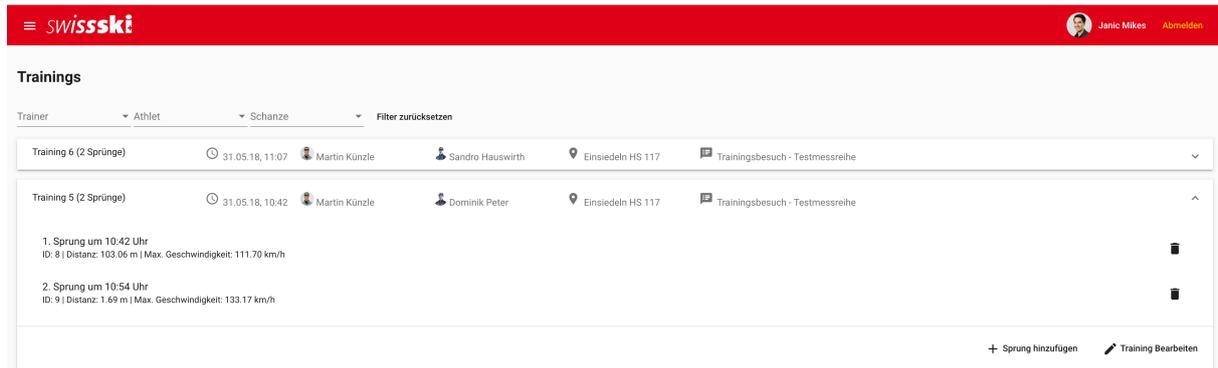


Abbildung 8.5: Aufgeklapptes Training in der Trainingsübersicht

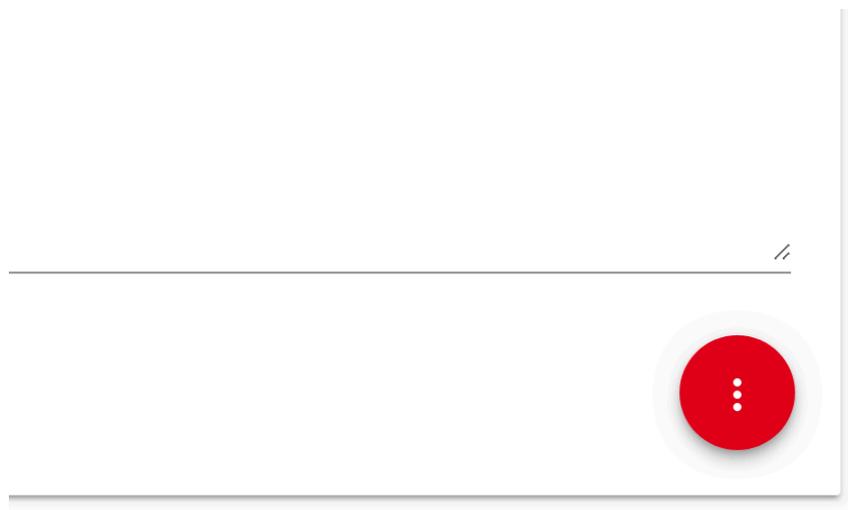


Abbildung 8.6: Der Action Button befindet sich unten rechts auf dem Bildschirm

## Training entfernen

Um ein Training zu entfernen muss man zuerst alle zugehörigen Sprünge entfernen. (Siehe Abschnitt 8.7.2)

In der Trainingsübersicht kann ein leeres Training aufgeklappt werden und es erscheint ein Button, um das Training zu entfernen.

## 8.7.2 Sprungverwaltung

### Jumpdashboard

Auf dem Jumpdashboard sieht man den 2D sowie den 3D Graphen, falls dem Sprung eine Messreihe zugewiesen wurde, sowie ein Video.

Über den Slider oder die Pfeiltasten auf der Tastatur kann man den aktuellen Zeitpunkt des Sprunges steuern.

Das Video kann durch anklicken des Bildes abgespielt und angehalten werden. Wenn das Video läuft, aktualisiert es den aktuellen Zeitpunkt des Sprunges, so dass die Graphen den-

selben Punkt hervorheben, welcher im Video gezeigt wird.

### Neuer Sprung hinzufügen

Ein neuer Sprung kann nur für ein bestehendes Training hinzugefügt werden. Dies geschieht entweder in der Trainingsübersicht bei aufgeklapptem Training (siehe Abbildung 8.5) oder vom Sprungdashboard aus über den Action Button (siehe Abbildung 8.6). Danach kann der Athlet, die Schanze sowie das Trainingsziel eingetragen werden und mit einem Klick auf *Training hinzufügen* gespeichert werden. Danach landet man auf dem Jumpdashboard des soeben erstellten Sprunges.

### Sprung bearbeiten

Im Jumpdashboard kann über den Action Button ein Sprung bearbeitet werden.

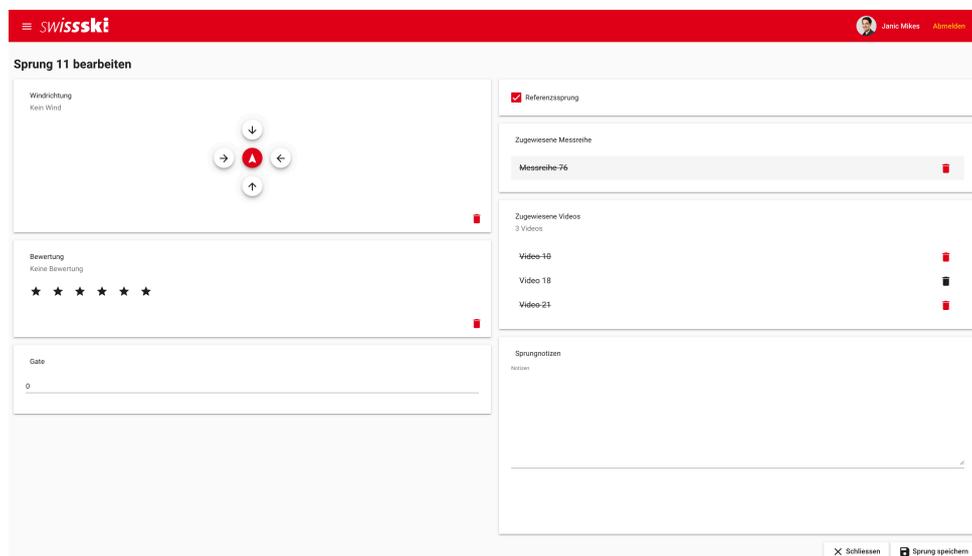


Abbildung 8.7: Sprung bearbeiten mit Trajektorie zum Löschen markiert

### Trajektorie hinzufügen

Auf dem Sprungdashboard kann über den Action Button eine Messreihe hinzugefügt werden.

Auf der linken Seite erscheinen alle Messreihen, welche auf dem Server verfügbar und noch keinem Sprung zugewiesen sind.

Nach dem Auswählen einer Messreihe wird diese als 2D Chart dargestellt. Es ist der Geschwindigkeits- sowie der Höhenverlauf zu sehen.

### Auswahl der vier Punkte

1. Messbeginn (Alles vor diesem Punkt wird nicht angezeigt)
2. Messende (Alles nach diesem Punkt wird nicht angezeigt)

3. Absprung (Wird benötigt für die Synchronisation und die Distanzbestimmung)
4. Landung (Wird benötigt für die Distanzbestimmung)

Mit einem Klick auf *Speichern & Schliessen* wird die Messreihe dem Sprung zugewiesen und man kehrt auf das Sprungdashboard zurück.

### **Trajektorie entfernen**

Siehe: Sprung bearbeiten 8.7.2.

Bei der angezeigten Messreihe auf den Abfalleimer klicken. Die Messreihe wird durchgestrichen dargestellt. Mit Speichern bestätigen.

### **Video hinzufügen**

Auf dem Sprungdashboard kann über den Action Button ein Video zum Sprung hinzugefügt werden.

Man klickt auf *Durchsuchen* und wählt die Videodatei auf dem Computer aus (Das Video muss im Format MP4 sein).

Nun kann das Video abgespielt werden. Pausiere das Video, wenn der Athlet den Tisch verlässt. Über die Tasten links und rechts neben dem Video kann der Absprungpunkt justiert werden.

Ist das Video beim Absprung angehalten, klickt man auf *Speichern & Hochladen*. Die aktuelle Stelle im Video wird als Absprungpunkt zusammen mit dem Video gespeichert und man kehrt auf das Sprungdashboard zurück.

### **Video entfernen**

Siehe: Sprung bearbeiten 8.7.2.

Bei den angezeigten Videos, welche gelöscht werden sollen, auf den Abfalleimer klicken. Die Videos werden durchgestrichen dargestellt. Mit Speichern bestätigen.

### **Sprünge vergleichen**

Auf dem Sprungdashboard kann über den Action Button ein Sprungvergleich gestartet werden.

Man vergleicht einen oder mehrere Referenzsprünge mit dem Sprung, von welchem der Vergleich gestartet wurde.

Um einen Sprung zu vergleichen, muss dieser auf derselben Schanze stattgefunden haben, und er muss als Referenzsprung gespeichert sein.

Nach dem Auswählen der Referenzsprünge erscheinen diese im Graphen.

## Sprung entfernen

In der Trainingsübersicht (siehe Abschnitt 8.7.1) ein Training aufklappen und bei einem Sprung auf den Abfalleimer klicken. Innerhalb von drei Sekunden erneut auf den Abfalleimer klicken, um das Löschen zu bestätigen.

## 8.8 Administratorenanleitung

Die Applikation wird über das Admin-WebUI administriert (siehe Abbildung 8.8). Es befindet sich unter <https://api.gps.skispringen.ch/admin/>. Hier folgt eine Übersicht über gängige Administratoren Tasks.

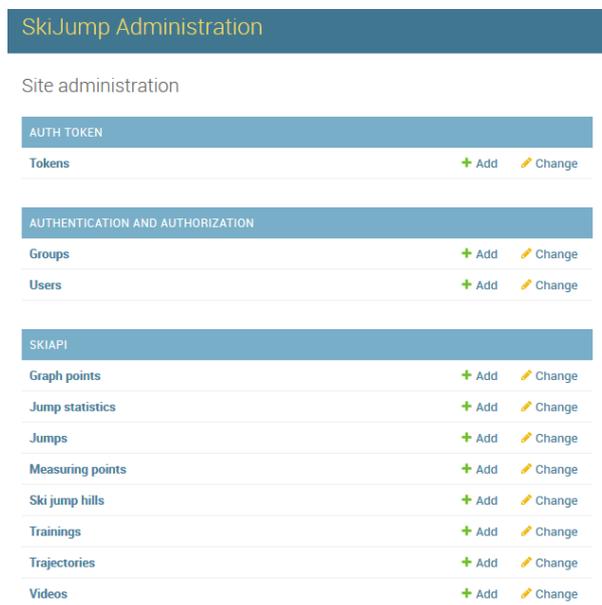


Abbildung 8.8: Admin UI Startseite

### 8.8.1 User verwalten

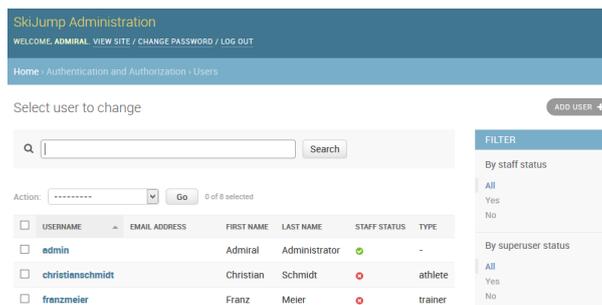


Abbildung 8.9: Admin UI Users

Menüpunkt: *Authentication and Authorization / Users*

Die Tabelle zeigt alle vorhandenen User an (siehe Abbildung 8.9). Ist bei Staff Status ein Haken, darf sich der User im Admin UI anmelden. Die Spalte *Type* bestimmt, ob ein User als Athlet oder als Trainer geführt wird. Wird kein *Type* angegeben, ist der User nicht in der Applikation ersichtlich.

### Neuer User erstellen

Auf den Knopf *ADD USER* klicken. Im ersten Schritt werden Username, Password, sowie der *Type* benötigt. Falls der User keinen *Type* hat, darf kein Avatar gewählt werden. Wird auf Save geklickt, können noch Vor- und Nachname sowie die E-Mail Adresse erfasst werden. Die Einstellungen *Groups* sowie *User permissions* haben keinen Einfluss auf die Berechtigungen innerhalb der Applikation.

### Passwort ändern

Auf den User klicken, dessen Passwort geändert werden soll. Unter *Password* befindet sich ein Link *you can change the password using this form*. Nach dem Klick auf diesen kann das Passwort des Users neu gesetzt werden.

### Berechtigungen anpassen

Auf den User klicken, dessen Berechtigungen geändert werden sollen. Auf *Athlete* oder *Trainer* setzen.

- *Athlete*: Ein Athlet kann alle seine Sprünge wie auch Referenzsprünge anschauen.
- *Trainer*: Ein Trainer sieht alle Sprünge, die auf dem System erfasst wurden.

## 8.8.2 Sprünge verwalten

Menüpunkt: *SkiApi / Jumps*

Es können dieselben Änderungen getätigt werden wie im User Interface.

## 8.8.3 Schanzen verwalten

Menüpunkt: *SkiApi / Ski jump hills*

Um eine Schanze erfassen zu können, werden folgende Informationen benötigt:

- *Name*: Der im User Interface anzuzeigende Name der Schanze
- *End of hill*: Die Koordinate des Schanzentisches (Absprungpunkt) im ECEF Format
- *Camera state side*: Kameraposition für die 3D Darstellung von der Seite
- *Camera state top*: Kameraposition für die 3D Darstellung von Oben

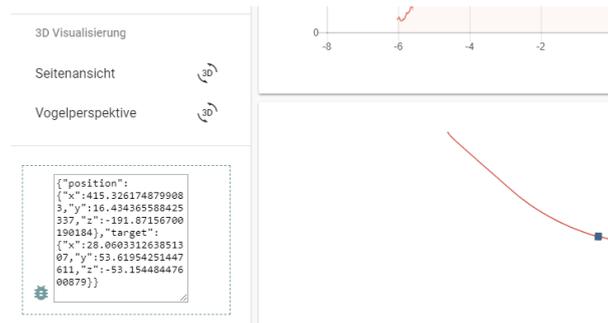


Abbildung 8.10: Kamera State

Die Kamerapositionen können aus dem User Interface ausgelesen werden. Dazu muss man sich als Administrator im User Interface einloggen und einen Sprung öffnen. Mit der linken Maustaste kann der Kamerawinkel verändert werden. Mit der rechten Maustaste die Position. Ist die Kameraposition zufriedenstellend, kann diese aus dem linken Menü (siehe Abbildung 8.10) kopiert und in die Schanze eingefügt werden.

#### 8.8.4 Videos verwalten

Menüpunkt: *SkiApi / Videos*

Ein einzelnes Video kann innerhalb der Videodetails (Klick auf den Filenamen) durch Klick auf den Delete Button gelöscht werden. Mehrere können in der Listenansicht über die Haken links vom Filenamen gewählt und durch Klicken auf *Action: Delete selected videos* gelöscht werden.

Innerhalb der Videodetails kann das Video auf einen anderen Punkt synchronisiert werden (Jump offset). Auch kann es einem anderen Jump zugewiesen werden.

Videos, welche über das Admin Interface gelöscht werden, sind nicht vom Filesystem des Servers gelöscht. Dies muss manuell erfolgen.

## 8.9 API Dokumentation

Mittels dem Projekt swagger2markup haben wir das von unserem Backend generierte Swaggerfile als PDF exportiert.

## Ski Jump API

## Table of Contents

1. Overview .....	1
1.1. Version information .....	1
1.2. URI scheme .....	1
1.3. Consumes .....	1
1.4. Produces .....	1
2. Paths .....	2
2.1. POST /auth/login/ .....	2
2.1.1. Parameters .....	2
2.1.2. Responses .....	2
2.1.3. Tags .....	2
2.2. GET /auth/profile/ .....	2
2.2.1. Responses .....	2
2.2.2. Tags .....	2
2.3. GET /jumphills/ .....	2
2.3.1. Responses .....	2
2.3.2. Tags .....	2
2.4. GET /jumphills/{id}/ .....	3
2.4.1. Parameters .....	3
2.4.2. Responses .....	3
2.4.3. Tags .....	3
2.5. GET /jumphills/{id}/jumps/ .....	3
2.5.1. Parameters .....	3
2.5.2. Responses .....	3
2.5.3. Tags .....	3
2.6. GET /jumphills/{id}/referencejumps/ .....	3
2.6.1. Parameters .....	3
2.6.2. Responses .....	4
2.6.3. Tags .....	4
2.7. POST /jumps/ .....	4
2.7.1. Parameters .....	4
2.7.2. Responses .....	4
2.7.3. Tags .....	4
2.8. GET /jumps/ .....	4
2.8.1. Responses .....	4
2.8.2. Tags .....	4
2.9. GET /jumps/latest/ .....	4
2.9.1. Responses .....	4
2.9.2. Tags .....	5

2.10. GET /jumps/{id}/	5
2.10.1. Parameters	5
2.10.2. Responses	5
2.10.3. Tags	5
2.11. DELETE /jumps/{id}/	5
2.11.1. Parameters	5
2.11.2. Responses	5
2.11.3. Tags	5
2.12. PATCH /jumps/{id}/	5
2.12.1. Parameters	5
2.12.2. Responses	6
2.12.3. Tags	6
2.13. GET /jumps/{id}/dashboard/	6
2.13.1. Parameters	6
2.13.2. Responses	6
2.13.3. Tags	6
2.14. GET /jumps/{id}/graph/	6
2.14.1. Parameters	6
2.14.2. Responses	6
2.14.3. Tags	7
2.15. GET /jumps/{id}/previewTrajectory/{trajectory_id}/	7
2.15.1. Parameters	7
2.15.2. Responses	7
2.15.3. Tags	7
2.16. GET /jumps/{id}/videos/	7
2.16.1. Parameters	7
2.16.2. Responses	7
2.16.3. Tags	8
2.17. POST /trainings/	8
2.17.1. Parameters	8
2.17.2. Responses	8
2.17.3. Tags	8
2.18. GET /trainings/	8
2.18.1. Responses	8
2.18.2. Tags	8
2.19. GET /trainings/{id}/	8
2.19.1. Parameters	8
2.19.2. Responses	8
2.19.3. Tags	9
2.20. PUT /trainings/{id}/	9
2.20.1. Parameters	9

2.20.2. Responses .....	9
2.20.3. Tags .....	9
2.21. DELETE /trainings/{id}/ .....	9
2.21.1. Parameters .....	9
2.21.2. Responses .....	9
2.21.3. Tags .....	9
2.22. PATCH /trainings/{id}/ .....	10
2.22.1. Parameters .....	10
2.22.2. Responses .....	10
2.22.3. Tags .....	10
2.23. POST /trajectories/ .....	10
2.23.1. Parameters .....	10
2.23.2. Responses .....	10
2.23.3. Tags .....	10
2.24. GET /trajectories/ .....	10
2.24.1. Responses .....	10
2.24.2. Tags .....	11
2.25. GET /trajectories/unassigned/ .....	11
2.25.1. Responses .....	11
2.25.2. Tags .....	11
2.26. GET /trajectories/{id}/ .....	11
2.26.1. Parameters .....	11
2.26.2. Responses .....	11
2.26.3. Tags .....	11
2.27. PUT /trajectories/{id}/ .....	11
2.27.1. Parameters .....	11
2.27.2. Responses .....	12
2.27.3. Tags .....	12
2.28. PATCH /trajectories/{id}/ .....	12
2.28.1. Parameters .....	12
2.28.2. Responses .....	12
2.28.3. Tags .....	12
2.29. GET /trajectories/{id}/points/ .....	12
2.29.1. Parameters .....	12
2.29.2. Responses .....	12
2.29.3. Tags .....	13
2.30. GET /users/ .....	13
2.30.1. Responses .....	13
2.30.2. Tags .....	13
2.31. GET /users/athletes/ .....	13
2.31.1. Responses .....	13

2.31.2. Tags .....	13
2.32. GET /users/trainers/ .....	13
2.32.1. Responses .....	13
2.32.2. Tags .....	13
2.33. GET /users/{id}/ .....	13
2.33.1. Parameters .....	13
2.33.2. Responses .....	14
2.33.3. Tags .....	14
2.34. POST /videos/ .....	14
2.34.1. Parameters .....	14
2.34.2. Responses .....	14
2.34.3. Tags .....	14
2.35. GET /videos/ .....	14
2.35.1. Responses .....	14
2.35.2. Tags .....	14
2.36. GET /videos/{id}/ .....	15
2.36.1. Parameters .....	15
2.36.2. Responses .....	15
2.36.3. Tags .....	15
2.37. PUT /videos/{id}/ .....	15
2.37.1. Parameters .....	15
2.37.2. Responses .....	15
2.37.3. Tags .....	15
2.38. DELETE /videos/{id}/ .....	15
2.38.1. Parameters .....	15
2.38.2. Responses .....	16
2.38.3. Tags .....	16
2.39. PATCH /videos/{id}/ .....	16
2.39.1. Parameters .....	16
2.39.2. Responses .....	16
2.39.3. Tags .....	16
3. Security .....	17
3.1. Basic .....	17
3.2. Bearer .....	17
4. Definitions .....	18
4.1. GraphPoint .....	18
4.2. JumpDashboard .....	18
4.3. JumpGraphModel .....	19
4.4. JumpListModel .....	19
4.5. JumpModel .....	20
4.6. JumpStatistic .....	20

4.7. MeasuringPoint .....	21
4.8. SkiJumpHill .....	21
4.9. TrainingList .....	21
4.10. TrainingModel .....	22
4.11. Trajectory.....	23
4.12. TrajectoryList .....	23
4.13. User .....	23
4.14. Video .....	24
4.15. _Token.....	24

# Chapter 1. Overview

## 1.1. Version information

*Version* : v1

## 1.2. URI scheme

*Host* : api-dev.gps.skispringen.ch

*BasePath* : /

*Schemes* : HTTPS

## 1.3. Consumes

- application/json

## 1.4. Produces

- application/json

## Chapter 2. Paths

### 2.1. POST /auth/login/

#### 2.1.1. Parameters

Type	Name	Schema
Query	<b>password</b> <i>required</i>	string
Query	<b>username</b> <i>required</i>	string

#### 2.1.2. Responses

HTTP Code	Schema
200	<a href="#">_Token</a>

#### 2.1.3. Tags

- auth

### 2.2. GET /auth/profile/

#### 2.2.1. Responses

HTTP Code	Schema
200	<a href="#">User</a>

#### 2.2.2. Tags

- auth

### 2.3. GET /jumphills/

#### 2.3.1. Responses

HTTP Code	Schema
200	<a href="#">&lt; SkiJumpHill &gt;</a> array

#### 2.3.2. Tags

- jumphills

## 2.4. GET /jumphills/{id}/

### 2.4.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this ski jump hill.	integer

### 2.4.2. Responses

HTTP Code	Schema
200	<a href="#">SkiJumpHill</a>

### 2.4.3. Tags

- jumphills

## 2.5. GET /jumphills/{id}/jumps/

### 2.5.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this ski jump hill.	integer

### 2.5.2. Responses

HTTP Code	Schema
200	< <a href="#">JumpListModel</a> > array

### 2.5.3. Tags

- jumphills

## 2.6. GET /jumphills/{id}/referencejumps/

### 2.6.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this ski jump hill.	integer

## 2.6.2. Responses

HTTP Code	Schema
200	< <a href="#">JumpListModel</a> > array

## 2.6.3. Tags

- jumphills

## 2.7. POST /jumps/

### 2.7.1. Parameters

Type	Name	Schema
Body	<b>data</b> <i>required</i>	<a href="#">JumpModel</a>

### 2.7.2. Responses

HTTP Code	Schema
201	<a href="#">JumpModel</a>

### 2.7.3. Tags

- jumps

## 2.8. GET /jumps/

### 2.8.1. Responses

HTTP Code	Schema
200	< <a href="#">JumpModel</a> > array

### 2.8.2. Tags

- jumps

## 2.9. GET /jumps/latest/

### 2.9.1. Responses

HTTP Code	Schema
200	<a href="#">JumpModel</a>

## 2.9.2. Tags

- jumps

## 2.10. GET /jumps/{id}/

### 2.10.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this jump.	integer

### 2.10.2. Responses

HTTP Code	Schema
200	<a href="#">JumpModel</a>

### 2.10.3. Tags

- jumps

## 2.11. DELETE /jumps/{id}/

### 2.11.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this jump.	integer

### 2.11.2. Responses

HTTP Code	Schema
204	No Content

### 2.11.3. Tags

- jumps

## 2.12. PATCH /jumps/{id}/

### 2.12.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this jump.	integer
Body	<b>data</b> <i>required</i>		<a href="#">JumpModel</a>

### 2.12.2. Responses

HTTP Code	Schema
200	<a href="#">JumpModel</a>

### 2.12.3. Tags

- jumps

## 2.13. GET /jumps/{id}/dashboard/

### 2.13.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this jump.	integer

### 2.13.2. Responses

HTTP Code	Schema
200	<a href="#">JumpDashboard</a>

### 2.13.3. Tags

- jumps

## 2.14. GET /jumps/{id}/graph/

### 2.14.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this jump.	integer

### 2.14.2. Responses

HTTP Code	Schema
200	< <a href="#">JumpGraphModel</a> > array

### 2.14.3. Tags

- jumps

## 2.15. GET

### /jumps/{id}/previewTrajectory/{trajectory\_id}/

#### 2.15.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this jump.	integer
Path	<b>trajectory_id</b> <i>required</i>		string

#### 2.15.2. Responses

HTTP Code	Schema
200	< <a href="#">GraphPoint</a> > array

### 2.15.3. Tags

- jumps

## 2.16. GET /jumps/{id}/videos/

#### 2.16.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this jump.	integer

#### 2.16.2. Responses

HTTP Code	Schema
200	< <a href="#">Video</a> > array

### 2.16.3. Tags

- jumps

## 2.17. POST /trainings/

### 2.17.1. Parameters

Type	Name	Schema
Body	<b>data</b> <i>required</i>	<a href="#">TrainingModel</a>

### 2.17.2. Responses

HTTP Code	Schema
201	<a href="#">TrainingModel</a>

### 2.17.3. Tags

- trainings

## 2.18. GET /trainings/

### 2.18.1. Responses

HTTP Code	Schema
200	< <a href="#">TrainingList</a> > array

### 2.18.2. Tags

- trainings

## 2.19. GET /trainings/{id}/

### 2.19.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this training.	integer

### 2.19.2. Responses

HTTP Code	Schema
200	<a href="#">TrainingModel</a>

### 2.19.3. Tags

- trainings

## 2.20. PUT /trainings/{id}/

### 2.20.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this training.	integer
Body	<b>data</b> <i>required</i>		<a href="#">TrainingModel</a>

### 2.20.2. Responses

HTTP Code	Schema
200	<a href="#">TrainingModel</a>

### 2.20.3. Tags

- trainings

## 2.21. DELETE /trainings/{id}/

### 2.21.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this training.	integer

### 2.21.2. Responses

HTTP Code	Schema
204	No Content

### 2.21.3. Tags

- trainings

## 2.22. PATCH /trainings/{id}/

### 2.22.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this training.	integer
Body	<b>data</b> <i>required</i>		<a href="#">TrainingModel</a>

### 2.22.2. Responses

HTTP Code	Schema
200	<a href="#">TrainingModel</a>

### 2.22.3. Tags

- trainings

## 2.23. POST /trajectories/

### 2.23.1. Parameters

Type	Name	Schema
Body	<b>data</b> <i>required</i>	<a href="#">Trajectory</a>

### 2.23.2. Responses

HTTP Code	Schema
201	<a href="#">Trajectory</a>

### 2.23.3. Tags

- trajectories

## 2.24. GET /trajectories/

### 2.24.1. Responses

HTTP Code	Schema
200	< <a href="#">TrajectoryList</a> > array

### 2.24.2. Tags

- trajectories

## 2.25. GET /trajectories/unassigned/

### 2.25.1. Responses

HTTP Code	Schema
200	< <a href="#">TrajectoryList</a> > array

### 2.25.2. Tags

- trajectories

## 2.26. GET /trajectories/{id}/

### 2.26.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this trajectory.	integer

### 2.26.2. Responses

HTTP Code	Schema
200	<a href="#">Trajectory</a>

### 2.26.3. Tags

- trajectories

## 2.27. PUT /trajectories/{id}/

### 2.27.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this trajectory.	integer
Body	<b>data</b> <i>required</i>		<a href="#">Trajectory</a>

### 2.27.2. Responses

HTTP Code	Schema
200	<a href="#">Trajectory</a>

### 2.27.3. Tags

- trajectories

## 2.28. PATCH /trajectories/{id}/

### 2.28.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this trajectory.	integer
Body	<b>data</b> <i>required</i>		<a href="#">Trajectory</a>

### 2.28.2. Responses

HTTP Code	Schema
200	<a href="#">Trajectory</a>

### 2.28.3. Tags

- trajectories

## 2.29. GET /trajectories/{id}/points/

### 2.29.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this trajectory.	integer

### 2.29.2. Responses

HTTP Code	Schema
200	< <a href="#">MeasuringPoint</a> > array

### 2.29.3. Tags

- trajectories

## 2.30. GET /users/

### 2.30.1. Responses

HTTP Code	Schema
200	< User > array

### 2.30.2. Tags

- users

## 2.31. GET /users/athletes/

### 2.31.1. Responses

HTTP Code	Schema
200	< User > array

### 2.31.2. Tags

- users

## 2.32. GET /users/trainers/

### 2.32.1. Responses

HTTP Code	Schema
200	< User > array

### 2.32.2. Tags

- users

## 2.33. GET /users/{id}/

### 2.33.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this user.	integer

### 2.33.2. Responses

HTTP Code	Schema
200	<a href="#">User</a>

### 2.33.3. Tags

- users

## 2.34. POST /videos/

### 2.34.1. Parameters

Type	Name	Schema
Body	<b>data</b> <i>required</i>	<a href="#">Video</a>

### 2.34.2. Responses

HTTP Code	Schema
201	<a href="#">Video</a>

### 2.34.3. Tags

- videos

## 2.35. GET /videos/

### 2.35.1. Responses

HTTP Code	Schema
200	< <a href="#">Video</a> > array

### 2.35.2. Tags

- videos

## 2.36. GET /videos/{id}/

### 2.36.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this video.	integer

### 2.36.2. Responses

HTTP Code	Schema
200	<a href="#">Video</a>

### 2.36.3. Tags

- videos

## 2.37. PUT /videos/{id}/

### 2.37.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this video.	integer
Body	<b>data</b> <i>required</i>		<a href="#">Video</a>

### 2.37.2. Responses

HTTP Code	Schema
200	<a href="#">Video</a>

### 2.37.3. Tags

- videos

## 2.38. DELETE /videos/{id}/

### 2.38.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this video.	integer

### 2.38.2. Responses

HTTP Code	Schema
204	No Content

### 2.38.3. Tags

- videos

## 2.39. PATCH /videos/{id}/

### 2.39.1. Parameters

Type	Name	Description	Schema
Path	<b>id</b> <i>required</i>	A unique integer value identifying this video.	integer
Body	<b>data</b> <i>required</i>		<a href="#">Video</a>

### 2.39.2. Responses

HTTP Code	Schema
200	<a href="#">Video</a>

### 2.39.3. Tags

- videos

## Chapter 3. Security

### 3.1. Basic

*Type* : basic

### 3.2. Bearer

*Type* : apiKey

*Name* : Authorization

*In* : HEADER

## Chapter 4. Definitions

### 4.1. GraphPoint

Name	Schema
<b>distance</b> <i>required</i>	number
<b>height</b> <i>required</i>	number
<b>measuring_point</b> <i>required</i>	integer
<b>time</b> <i>required</i>	number
<b>velocity</b> <i>required</i>	number
<b>x</b> <i>required</i>	number
<b>y</b> <i>required</i>	number
<b>z</b> <i>required</i>	number

### 4.2. JumpDashboard

Name	Description	Schema
<b>date</b> <i>optional</i>		string(date-time)
<b>gate</b> <i>optional</i>	<b>Minimum value</b> : -2147483648 <b>Maximum value</b> : 2147483647	integer
<b>grade</b> <i>optional</i>	<b>Minimum value</b> : -32768 <b>Maximum value</b> : 32767	integer
<b>graphpoints</b> <i>required</i>		< <a href="#">GraphPoint</a> > array
<b>id</b> <i>optional</i> <i>read-only</i>		integer
<b>jump_statistic</b> <i>required</i>		<a href="#">JumpStatistic</a>

Name	Description	Schema
<b>landing</b> <i>required</i>		<a href="#">GraphPoint</a>
<b>notes</b> <i>optional</i>		string
<b>takeoff</b> <i>required</i>		<a href="#">GraphPoint</a>
<b>training</b> <i>required</i>		<a href="#">TrainingModel</a>
<b>trajectory</b> <i>optional</i>		integer
<b>videos</b> <i>required</i>		< <a href="#">Video</a> > array
<b>wind_direction</b> <i>optional</i>		integer

### 4.3. JumpGraphModel

Name	Schema
<b>graphpoints</b> <i>required</i>	< <a href="#">GraphPoint</a> > array
<b>id</b> <i>optional</i> <i>read-only</i>	integer
<b>landing</b> <i>required</i>	<a href="#">GraphPoint</a>
<b>takeoff</b> <i>required</i>	<a href="#">GraphPoint</a>

### 4.4. JumpListModel

Name	Schema
<b>date</b> <i>optional</i>	string(date-time)
<b>id</b> <i>optional</i> <i>read-only</i>	integer
<b>jump_statistic</b> <i>required</i>	<a href="#">JumpStatistic</a>

Name	Schema
<b>training</b> <i>required</i>	integer

## 4.5. JumpModel

Name	Description	Schema
<b>date</b> <i>optional</i>		string(date-time)
<b>gate</b> <i>optional</i>	<b>Minimum value : -2147483648</b> <b>Maximum value : 2147483647</b>	integer
<b>grade</b> <i>optional</i>	<b>Minimum value : -32768</b> <b>Maximum value : 32767</b>	integer
<b>id</b> <i>optional</i> <i>read-only</i>		integer
<b>notes</b> <i>optional</i>		string
<b>reference_jump</b> <i>optional</i>		boolean
<b>training</b> <i>required</i>		integer
<b>trajectory</b> <i>optional</i>		integer
<b>videos</b> <i>optional</i>		< Video > array
<b>wind_direction</b> <i>optional</i>		integer

## 4.6. JumpStatistic

Name	Schema
<b>air_time</b> <i>required</i>	number
<b>centric_offset</b> <i>optional</i>	number
<b>jump_distance</b> <i>required</i>	number

Name	Schema
<b>jump_height</b> <i>optional</i>	number
<b>landing_velocity</b> <i>required</i>	number
<b>max_velocity</b> <i>required</i>	number
<b>takeoff_velocity</b> <i>required</i>	number

## 4.7. MeasuringPoint

Name	Schema
<b>gps_timestamp</b> <i>required</i>	string(date-time)
<b>id</b> <i>optional</i> <i>read-only</i>	integer
<b>point</b> <i>required</i>	string

## 4.8. SkiJumpHill

Name	Description	Schema
<b>camera_state_side</b> <i>optional</i>	Minimum length : 1	string
<b>camera_state_top</b> <i>optional</i>	Minimum length : 1	string
<b>id</b> <i>optional</i> <i>read-only</i>		integer
<b>name</b> <i>required</i>	Minimum length : 1	string

## 4.9. TrainingList

Name	Schema
<b>athlete</b> <i>required</i>	User

Name	Schema
<b>goal</b> <i>optional</i>	string
<b>id</b> <i>optional</i> <i>read-only</i>	integer
<b>jump_hill_name</b> <i>optional</i> <i>read-only</i>	string
<b>jumps</b> <i>optional</i> <i>read-only</i>	string
<b>trainer</b> <i>required</i>	<a href="#">User</a>
<b>trainer_id</b> <i>optional</i> <i>read-only</i>	string

## 4.10. TrainingModel

Name	Schema
<b>athlete</b> <i>optional</i>	<a href="#">User</a>
<b>athlete_id</b> <i>optional</i>	integer
<b>goal</b> <i>optional</i>	string
<b>id</b> <i>optional</i> <i>read-only</i>	integer
<b>jump_hill</b> <i>optional</i>	<a href="#">SkiJumpHill</a>
<b>jump_hill_id</b> <i>optional</i>	integer
<b>trainer</b> <i>optional</i>	<a href="#">User</a>
<b>trainer_id</b> <i>optional</i>	integer

## 4.11. Trajectory

Name	Schema
<b>begin</b> <i>optional</i>	integer
<b>end</b> <i>optional</i>	integer
<b>id</b> <i>optional</i> <i>read-only</i>	integer
<b>landing</b> <i>optional</i>	integer
<b>measuringpoints</b> <i>required</i>	< <a href="#">MeasuringPoint</a> > array
<b>takeoff</b> <i>optional</i>	integer

## 4.12. TrajectoryList

Name	Schema
<b>begin</b> <i>optional</i>	integer
<b>date</b> <i>optional</i> <i>read-only</i>	string
<b>end</b> <i>optional</i>	integer
<b>id</b> <i>optional</i> <i>read-only</i>	integer
<b>landing</b> <i>optional</i>	integer
<b>takeoff</b> <i>optional</i>	integer

## 4.13. User

Name	Description	Schema
<b>avatar</b> <i>optional</i> <i>read-only</i>		string

Name	Description	Schema
<b>email</b> <i>optional</i>	Maximal length : 254	string(email)
<b>first_name</b> <i>optional</i>	Maximal length : 30	string
<b>groups</b> <i>optional</i>	The groups this user belongs to. A user will get all permissions granted to each of their groups.	< integer > array
<b>id</b> <i>optional</i> <i>read-only</i>		integer
<b>last_name</b> <i>optional</i>	Maximal length : 150	string
<b>type</b> <i>optional</i> <i>read-only</i>		string
<b>username</b> <i>required</i>	Required. 150 characters or fewer. Letters, digits and @/./-/_ only. + **Length** : `1 - 150` + **Pattern** : `^[\\w.@-]+\$`	string

#### 4.14. Video

Name	Schema
<b>file_location</b> <i>required</i> <i>read-only</i>	string(uri)
<b>fps</b> <i>optional</i>	number
<b>id</b> <i>optional</i> <i>read-only</i>	integer
<b>jump</b> <i>optional</i>	integer
<b>jump_offset</b> <i>optional</i>	number

#### 4.15. \_Token

Name	Description	Schema
<b>token</b> <i>required</i>	Minimum length : 1	string

## 8.10 NPM Audit

```
1 === npm audit security report ===
2
3 # Run npm install --save-dev protractor@5.3.2 to resolve 5 vulnerabilities
4
5 High          Denial of Service
6
7 Package       https-proxy-agent
8
9 Dependency of  protractor [dev]
10
11 Path          protractor > saucelabs > https-proxy-agent
12
13 More info     https://nodesecurity.io/advisories/593
14
15
16
17
18 Moderate     Prototype pollution
19
20 Package       hoek
21
22 Dependency of  protractor [dev]
23
24 Path          protractor > webdriver-manager > request > hawk > boom >
25               hoek
26
27 More info     https://nodesecurity.io/advisories/566
28
29
30
31
32 Moderate     Prototype pollution
33
34 Package       hoek
35
36 Dependency of  protractor [dev]
37
38 Path          protractor > webdriver-manager > request > hawk > cryptiles
39               > boom > hoek
40
41 More info     https://nodesecurity.io/advisories/566
42
43
44
45
46 Moderate     Prototype pollution
47
48 Package       hoek
49
```

```
50 Dependency of   protractor [dev]
51
52 Path           protractor > webdriver-manager > request > hawk > hoek
53
54 More info      https://nodesecurity.io/advisories/566
55
56
57
58
59 Moderate      Prototype pollution
60
61 Package       hoek
62
63 Dependency of   protractor [dev]
64
65 Path           protractor > webdriver-manager > request > hawk > sntp >
66               hoek
67
68 More info      https://nodesecurity.io/advisories/566
69
70
71
72 # Run  npm update request --depth 5  to resolve 8 vulnerabilities
73
74 Moderate      Prototype pollution
75
76 Package       hoek
77
78 Dependency of   karma [dev]
79
80 Path           karma > log4js > hipchat-notifier > request > hawk > boom >
81               hoek
82
83 More info      https://nodesecurity.io/advisories/566
84
85
86
87
88 Moderate      Prototype pollution
89
90 Package       hoek
91
92 Dependency of   karma [dev]
93
94 Path           karma > log4js > hipchat-notifier > request > hawk >
95               cryptiles > boom > hoek
96
97 More info      https://nodesecurity.io/advisories/566
98
99
100
101
```

102	Moderate	Prototype pollution
103		
104	Package	hoek
105		
106	Dependency of	karma [dev]
107		
108	Path	karma > log4js > hipchat-notifier > request > hawk > hoek
109		
110	More info	<a href="https://nodesecurity.io/advisories/566">https://nodesecurity.io/advisories/566</a>
111		
112		
113		
114		
115	Moderate	Prototype pollution
116		
117	Package	hoek
118		
119	Dependency of	karma [dev]
120		
121	Path	karma > log4js > hipchat-notifier > request > hawk > sntp > hoek
122		
123		
124	More info	<a href="https://nodesecurity.io/advisories/566">https://nodesecurity.io/advisories/566</a>
125		
126		
127		
128		
129	Moderate	Prototype pollution
130		
131	Package	hoek
132		
133	Dependency of	karma [dev]
134		
135	Path	karma > log4js > slack-node > requestretry > request > hawk > boom > hoek
136		
137		
138	More info	<a href="https://nodesecurity.io/advisories/566">https://nodesecurity.io/advisories/566</a>
139		
140		
141		
142		
143	Moderate	Prototype pollution
144		
145	Package	hoek
146		
147	Dependency of	karma [dev]
148		
149	Path	karma > log4js > slack-node > requestretry > request > hawk > cryptiles > boom > hoek
150		
151		
152	More info	<a href="https://nodesecurity.io/advisories/566">https://nodesecurity.io/advisories/566</a>
153		

154		
155		
156		
157	Moderate	Prototype pollution
158		
159	Package	hoek
160		
161	Dependency of	karma [dev]
162		
163	Path	karma > log4js > slack-node > requestretry > request > hawk
164		> hoek
165		
166	More info	<a href="https://nodesecurity.io/advisories/566">https://nodesecurity.io/advisories/566</a>
167		
168		
169		
170		
171	Moderate	Prototype pollution
172		
173	Package	hoek
174		
175	Dependency of	karma [dev]
176		
177	Path	karma > log4js > slack-node > requestretry > request > hawk
178		> sntp > hoek
179		
180	More info	<a href="https://nodesecurity.io/advisories/566">https://nodesecurity.io/advisories/566</a>
181		
182		
183		
184		
185		Manual Review
186		Some vulnerabilities require your attention to resolve
187		
188		Visit <a href="https://go.npm.me/audit-guide">https://go.npm.me/audit-guide</a> for additional guidance
189		
190		
191	Moderate	Prototype pollution
192		
193	Package	hoek
194		
195	Patched in	> 4.2.0 < 5.0.0    >= 5.0.3
196		
197	Dependency of	karma [dev]
198		
199	Path	karma > log4js > loggly > request > hawk > boom > hoek
200		
201	More info	<a href="https://nodesecurity.io/advisories/566">https://nodesecurity.io/advisories/566</a>
202		
203		
204	Moderate	Prototype pollution
205		

206	Package	hoek
207		
208	Patched in	> 4.2.0 < 5.0.0    >= 5.0.3
209		
210	Dependency of	karma [dev]
211		
212	Path	karma > log4js > loggly > request > hawk > cryptiles > boom
213		> hoek
214		
215	More info	<a href="https://nodesecurity.io/advisories/566">https://nodesecurity.io/advisories/566</a>
216		
217		
218	Moderate	Prototype pollution
219		
220	Package	hoek
221		
222	Patched in	> 4.2.0 < 5.0.0    >= 5.0.3
223		
224	Dependency of	karma [dev]
225		
226	Path	karma > log4js > loggly > request > hawk > hoek
227		
228	More info	<a href="https://nodesecurity.io/advisories/566">https://nodesecurity.io/advisories/566</a>
229		
230		
231	Moderate	Prototype pollution
232		
233	Package	hoek
234		
235	Patched in	> 4.2.0 < 5.0.0    >= 5.0.3
236		
237	Dependency of	karma [dev]
238		
239	Path	karma > log4js > loggly > request > hawk > sntp > hoek
240		
241	More info	<a href="https://nodesecurity.io/advisories/566">https://nodesecurity.io/advisories/566</a>
242		
243		
244	Moderate	Memory Exposure
245		
246	Package	tunnel-agent
247		
248	Patched in	>=0.6.0
249		
250	Dependency of	karma [dev]
251		
252	Path	karma > log4js > loggly > request > tunnel-agent
253		
254	More info	<a href="https://nodesecurity.io/advisories/598">https://nodesecurity.io/advisories/598</a>
255		
256		
257	Low	Regular Expression Denial of Service

```
258 Package          timespan
259
260 Patched in       No patch available
261
262 Dependency of    karma [dev]
263
264 Path            karma > log4js > loggly > timespan
265
266 More info       https://nodesecurity.io/advisories/533
267
268
269 found 19 vulnerabilities (1 low, 17 moderate, 1 high) in 7615 scanned packages
270 run `npm audit fix` to fix 13 of them.
271 6 vulnerabilities require manual review. See the full report for details.
```

Listing 8.25: NPM Audit Resultate

## 8.11 Frontend Thirdparty Lizenzen

```
1 moment@2.22.1
2 MIT
3 Copyright (c) JS Foundation and other contributors
4
5 Permission is hereby granted, free of charge, to any person
6 obtaining a copy of this software and associated documentation
7 files (the "Software"), to deal in the Software without
8 restriction, including without limitation the rights to use,
9 copy, modify, merge, publish, distribute, sublicense, and/or sell
10 copies of the Software, and to permit persons to whom the
11 Software is furnished to do so, subject to the following
12 conditions:
13
14 The above copyright notice and this permission notice shall be
15 included in all copies or substantial portions of the Software.
16
17 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
18 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES
19 OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
20 NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT
21 HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY,
22 WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING
23 FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR
24 OTHER DEALINGS IN THE SOFTWARE.
25
26 cache-loader@1.2.2
27 MIT
28 Copyright JS Foundation and other contributors
29
30 Permission is hereby granted, free of charge, to any person obtaining
31 a copy of this software and associated documentation files (the
```

```
32 'Software'), to deal in the Software without restriction, including
33 without limitation the rights to use, copy, modify, merge, publish,
34 distribute, sublicense, and/or sell copies of the Software, and to
35 permit persons to whom the Software is furnished to do so, subject to
36 the following conditions:
37
38 The above copyright notice and this permission notice shall be
39 included in all copies or substantial portions of the Software.
40
41 THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
42 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
43 MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
44 IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
45 CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
46 TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
47 SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
48
49 @angular-devkit/build-optimizer@0.3.2
50 MIT
51 The MIT License
52
53 Copyright (c) 2017 Google, Inc.
54
55 Permission is hereby granted, free of charge, to any person obtaining a copy
56 of this software and associated documentation files (the "Software"), to deal
57 in the Software without restriction, including without limitation the rights
58 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
59 copies of the Software, and to permit persons to whom the Software is
60 furnished to do so, subject to the following conditions:
61
62 The above copyright notice and this permission notice shall be included in all
63 copies or substantial portions of the Software.
64
65 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
66 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
67 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
68 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
69 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
70 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
71 SOFTWARE.
72
73 chart.js@2.7.2
74 MIT
75 The MIT License (MIT)
76
77 Copyright (c) 2018 Chart.js Contributors
78
79 Permission is hereby granted, free of charge, to any person obtaining a copy of
  ↳ this software and associated documentation files (the "Software"), to deal in
  ↳ the Software without restriction, including without limitation the rights to
  ↳ use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
  ↳ of the Software, and to permit persons to whom the Software is furnished to do
```

↪ so, subject to the following conditions:

80

81 The above copyright notice and this permission notice shall be included in all  
↪ copies or substantial portions of the Software.

82

83 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED  
↪ , INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR  
↪ A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR  
↪ COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER  
↪ IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN  
↪ CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

84

85 chartjs-plugin-annotation@0.5.7

86 MIT

87 The MIT License (MIT)

88 Copyright (c) 2016 Evert Timberg

89

90 Permission is hereby granted, free of charge, to any person obtaining a copy of  
↪ this software and associated documentation files (the "Software"), to deal in  
↪ the Software without restriction, including without limitation the rights to  
↪ use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies  
↪ of the Software, and to permit persons to whom the Software is furnished to do  
↪ so, subject to the following conditions:

91

92 The above copyright notice and this permission notice shall be included in all  
↪ copies or substantial portions of the Software.

93

94 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED  
↪ , INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR  
↪ A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR  
↪ COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER  
↪ IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN  
↪ CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

95

96 three@0.91.0

97 MIT

98 The MIT License

99

100 Copyright © 2010-2018 three.js authors

101

102 Permission is hereby granted, free of charge, to any person obtaining a copy  
103 of this software and associated documentation files (the "Software"), to deal  
104 in the Software without restriction, including without limitation the rights  
105 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
106 copies of the Software, and to permit persons to whom the Software is  
107 furnished to do so, subject to the following conditions:

108

109 The above copyright notice and this permission notice shall be included in  
110 all copies or substantial portions of the Software.

111

112 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
113 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,

```
114 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
115 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
116 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
117 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
118 THE SOFTWARE.
119
120 webpack@3.11.0
121 MIT
122 Copyright JS Foundation and other contributors
123
124 Permission is hereby granted, free of charge, to any person obtaining
125 a copy of this software and associated documentation files (the
126 'Software'), to deal in the Software without restriction, including
127 without limitation the rights to use, copy, modify, merge, publish,
128 distribute, sublicense, and/or sell copies of the Software, and to
129 permit persons to whom the Software is furnished to do so, subject to
130 the following conditions:
131
132 The above copyright notice and this permission notice shall be
133 included in all copies or substantial portions of the Software.
134
135 THE SOFTWARE IS PROVIDED 'AS IS', WITHOUT WARRANTY OF ANY KIND,
136 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
137 MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
138 IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
139 CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
140 TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
141 SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
142
143 color-name@1.1.3
144 MIT
145 The MIT License (MIT)
146 Copyright (c) 2015 Dmitry Ivanov
147
148 Permission is hereby granted, free of charge, to any person obtaining a copy of
149 ↪ this software and associated documentation files (the "Software"), to deal in
150 ↪ the Software without restriction, including without limitation the rights to
151 ↪ use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies
152 ↪ of the Software, and to permit persons to whom the Software is furnished to do
153 ↪ so, subject to the following conditions:
154
155 The above copyright notice and this permission notice shall be included in all
156 ↪ copies or substantial portions of the Software.
157
158 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED
159 ↪ , INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR
160 ↪ A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR
161 ↪ COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER
162 ↪ IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN
163 ↪ CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
```

```
155 MIT
156 Copyright (c) 2012 Heather Arthur
157
158 Permission is hereby granted, free of charge, to any person obtaining
159 a copy of this software and associated documentation files (the
160 "Software"), to deal in the Software without restriction, including
161 without limitation the rights to use, copy, modify, merge, publish,
162 distribute, sublicense, and/or sell copies of the Software, and to
163 permit persons to whom the Software is furnished to do so, subject to
164 the following conditions:
165
166 The above copyright notice and this permission notice shall be
167 included in all copies or substantial portions of the Software.
168
169 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
170 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
171 MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
172 NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
173 LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
174 OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
175 WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
176
177 @angular/core@5.2.10
178 MIT
179 MIT
180
181 chartjs-color-string@0.5.0
182 MIT
183 Copyright (c) 2011 Heather Arthur <fayearthur@gmail.com>
184
185 Permission is hereby granted, free of charge, to any person obtaining
186 a copy of this software and associated documentation files (the
187 "Software"), to deal in the Software without restriction, including
188 without limitation the rights to use, copy, modify, merge, publish,
189 distribute, sublicense, and/or sell copies of the Software, and to
190 permit persons to whom the Software is furnished to do so, subject to
191 the following conditions:
192
193 The above copyright notice and this permission notice shall be
194 included in all copies or substantial portions of the Software.
195
196 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
197 EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
198 MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
199 NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE
200 LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION
201 OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION
202 WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
203
204 chartjs-plugin-zoom@0.6.3
205 MIT
206 The MIT License (MIT)
```

207 Copyright (c) 2013-2016 Nick Downie  
208  
209 Permission is hereby granted, free of charge, to any person obtaining a copy of  
↪ this software and associated documentation files (the "Software"), to deal in  
↪ the Software without restriction, including without limitation the rights to  
↪ use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies  
↪ of the Software, and to permit persons to whom the Software is furnished to do  
↪ so, subject to the following conditions:  
210  
211 The above copyright notice and this permission notice shall be included in all  
↪ copies or substantial portions of the Software.  
212  
213 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED  
↪ , INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR  
↪ A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR  
↪ COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER  
↪ IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN  
↪ CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.  
214  
215 hammerjs@2.0.8  
216 MIT  
217 The MIT License (MIT)  
218  
219 Copyright (C) 2011-2014 by Jorik Tangelder (Eight Media)  
220  
221 Permission is hereby granted, free of charge, to any person obtaining a copy  
222 of this software and associated documentation files (the "Software"), to deal  
223 in the Software without restriction, including without limitation the rights  
224 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
225 copies of the Software, and to permit persons to whom the Software is  
226 furnished to do so, subject to the following conditions:  
227  
228 The above copyright notice and this permission notice shall be included in  
229 all copies or substantial portions of the Software.  
230  
231 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
232 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
233 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE  
234 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
235 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
236 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN  
237 THE SOFTWARE.  
238  
239 @angular/common@5.2.10  
240 MIT  
241 MIT  
242  
243 @angular/cdk@5.2.5  
244 MIT  
245 The MIT License  
246  
247 Copyright (c) 2018 Google LLC.

248  
249 Permission is hereby granted, free of charge, to any person obtaining a copy  
250 of this software and associated documentation files (the "Software"), to deal  
251 in the Software without restriction, including without limitation the rights  
252 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
253 copies of the Software, and to permit persons to whom the Software is  
254 furnished to do so, subject to the following conditions:  
255  
256 The above copyright notice and this permission notice shall be included in  
257 all copies or substantial portions of the Software.  
258  
259 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
260 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
261 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE  
262 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
263 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
264 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN  
265 THE SOFTWARE.  
266  
267 @angular/platform-browser@5.2.10  
268 MIT  
269 MIT  
270  
271 @angular/router@5.2.10  
272 MIT  
273 MIT  
274  
275 @angular/material@5.2.5  
276 MIT  
277 The MIT License  
278  
279 Copyright (c) 2018 Google LLC.  
280  
281 Permission is hereby granted, free of charge, to any person obtaining a copy  
282 of this software and associated documentation files (the "Software"), to deal  
283 in the Software without restriction, including without limitation the rights  
284 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
285 copies of the Software, and to permit persons to whom the Software is  
286 furnished to do so, subject to the following conditions:  
287  
288 The above copyright notice and this permission notice shall be included in  
289 all copies or substantial portions of the Software.  
290  
291 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
292 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
293 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE  
294 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
295 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
296 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN  
297 THE SOFTWARE.  
298  
299 @angular/forms@5.2.10

```
300 MIT
301 MIT
302
303 @angular/animations@5.2.10
304 MIT
305 MIT
306
307 @angular/flex-layout@5.0.0-beta.13
308 MIT
309 The MIT License
310
311 Copyright (c) 2018 Google LLC.
312
313 Permission is hereby granted, free of charge, to any person obtaining a copy
314 of this software and associated documentation files (the "Software"), to deal
315 in the Software without restriction, including without limitation the rights
316 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
317 copies of the Software, and to permit persons to whom the Software is
318 furnished to do so, subject to the following conditions:
319
320 The above copyright notice and this permission notice shall be included in
321 all copies or substantial portions of the Software.
322
323 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
324 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
325 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
326 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
327 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
328 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN
329 THE SOFTWARE.
330
331 @types/three@0.91.10
332 MIT
333 MIT License
334
335 Copyright (c) Microsoft Corporation. All rights reserved.
336
337 Permission is hereby granted, free of charge, to any person obtaining a copy
338 of this software and associated documentation files (the "Software"), to deal
339 in the Software without restriction, including without limitation the rights
340 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
341 copies of the Software, and to permit persons to whom the Software is
342 furnished to do so, subject to the following conditions:
343
344 The above copyright notice and this permission notice shall be included in all
345 copies or substantial portions of the Software.
346
347 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
348 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
349 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
350 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
351 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
```

352 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE  
353 SOFTWARE

354  
355 @angular/platform-browser-dynamic@5.2.10

356 MIT

357 MIT

358

359 core-js@2.5.5

360 MIT

361 Copyright (c) 2014-2018 Denis Pushkarev

362

363 Permission is hereby granted, free of charge, to any person obtaining a copy  
364 of this software and associated documentation files (the "Software"), to deal  
365 in the Software without restriction, including without limitation the rights  
366 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
367 copies of the Software, and to permit persons to whom the Software is  
368 furnished to do so, subject to the following conditions:

369

370 The above copyright notice and this permission notice shall be included in  
371 all copies or substantial portions of the Software.

372

373 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
374 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
375 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE  
376 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
377 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
378 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN  
379 THE SOFTWARE.

380

381 zone.js@0.8.26

382 MIT

383 The MIT License

384

385 Copyright (c) 2016-2018 Google, Inc.

386

387 Permission is hereby granted, free of charge, to any person obtaining a copy  
388 of this software and associated documentation files (the "Software"), to deal  
389 in the Software without restriction, including without limitation the rights  
390 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell  
391 copies of the Software, and to permit persons to whom the Software is  
392 furnished to do so, subject to the following conditions:

393

394 The above copyright notice and this permission notice shall be included in  
395 all copies or substantial portions of the Software.

396

397 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR  
398 IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,  
399 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE  
400 AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER  
401 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,  
402 OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN  
403 THE SOFTWARE.

## 8.12 PIL License

Siehe Abschnitt 8.26.

```
1 The Python Imaging Library (PIL) is
2
3     Copyright © 1997-2011 by Secret Labs AB
4     Copyright © 1995-2011 by Fredrik Lundh
5
6 By obtaining, using, and/or copying this software and/or its associated
7     ↪ documentation, you agree that you have read, understood, and will comply with
8     ↪ the following terms and conditions:
9
10 Permission to use, copy, modify, and distribute this software and its associated
11     ↪ documentation for any purpose and without fee is hereby granted, provided that
12     ↪ the above copyright notice appears in all copies, and that both that
13     ↪ copyright notice and this permission notice appear in supporting documentation
14     ↪ , and that the name of Secret Labs AB or the author not be used in advertising
15     ↪ or publicity pertaining to distribution of the software without specific,
16     ↪ written prior permission.
17
18 SECRET LABS AB AND THE AUTHOR DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS
19     ↪ SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN
20     ↪ NO EVENT SHALL SECRET LABS AB OR THE AUTHOR BE LIABLE FOR ANY SPECIAL,
21     ↪ INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM
22     ↪ LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR
23     ↪ OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
24     ↪ PERFORMANCE OF THIS SOFTWARE.
```

Listing 8.26: PIL Lizenz