RHEINISCHE FACHHOCHSCHULE KÖLN

University of Applied Sciences

Fachbereich: Wirtschaft & Recht

Studiengang: Business Information Management (B.Sc.)



**Web & Mobile Development**

**Projektdokumentation**

Webanwendung

Sportfreunde RFH (Laufseite)

Vorgelegt von: Thorsten Müller

212141002

Patrick Piechatzek

212141001

Sebastian Röder

212141008

Prüfer: Patric Steffen

Sommersemester 2015

Inhalt

[1. Einleitung 3](#_Toc419695664)

[1.1. Motivation 3](#_Toc419695665)

[1.2. Inhalt 3](#_Toc419695666)

[1.3. Ziele der Projektarbeit 4](#_Toc419695667)

[2. Vorgehensweise 5](#_Toc419695668)

[2.1. Organisation der Projektarbeit 5](#_Toc419695669)

[2.2. Aufgaben und Arbeitsschritte 5](#_Toc419695670)

[2.3. Verteilung der Aufgaben auf die Gruppenmitglieder 6](#_Toc419695671)

[3. Anforderungsanalyse 7](#_Toc419695672)

[3.1. Funktionale Anforderungen 7](#_Toc419695673)

[3.2. Nichtfunktionale Anforderungen 9](#_Toc419695674)

[4. Technische Umsetzung 9](#_Toc419695675)

[4.1. Eingesetzte Anwendungen 9](#_Toc419695676)

[4.2. Aufgaben der Anforderungen 10](#_Toc419695677)

[4.3. Abläufe innerhalb der Software 10](#_Toc419695678)

[4.4. Strukturelemente der Software 12](#_Toc419695679)

[4.5. Datenmodell-Entwurf 13](#_Toc419695680)

[5. Eingereichte Ergebnisse 13](#_Toc419695681)

[6. Installationsanleitung 17](#_Toc419695682)

[7. Fazit 18](#_Toc419695683)

[8. Quellenverzeichnis 18](#_Toc419695684)

# Einleitung

Die vorliegende Projektarbeit befasst sich mit der Planung, Realisierung und Dokumentation einer sogenannten „Single-Page“ Webseitenanwendung.

## Motivation

Diese Dokumentation betrachtet die Durchführung eines Entwicklungsprojekts im Bereich Web- & Mobile-Development im Rahmen des berufsbegleitenden Studiengangs Wirtschaftsinformatik an der Rheinischen Fachhochschule Köln.

Durch den direkten Umgang mit den notwendigen Programmiersprachen und Frameworks, wird so die Handhabung und Nutzung der zur Verfügung stehenden Ressourcen ausgeschöpft und in der Gruppenarbeit zum Ergebnis gebracht.

Um das Projekt für alle Beteiligten gleichermaßen interessant zu gestalten, wurden sämtliche Interessen der Projektteilnehmer berücksichtigt. Hieraus entstand die Idee eine Webanwendung zu entwickeln, die laufbegeisterten Sportlern als Plattform zur Verfügung steht.

## Inhalt

Die entstandene Webanwendung umfasst eine Desktop sowie eine mobile Version. Die Anforderungen der Desktopversion der Seite sind in ihrer Funktionalität und ihrem Aussehen komplexer gestellt als die der mobilen Version.

Die mobile Version spiegelt eine für mobile Endgeräte angepasste Seite wieder, die auf dem JavaScript Framework *Sencha Touch* basiert. Sie zeigt auf ihrer Startseite die gesamte Liste aller Läufe und bietet die Möglichkeit einer Detailansicht der einzelnen Einträge.

Für die umfangreichere Desktopversion bestehen eine Benutzerauthentifizierung, die Listen- und Detailansicht und außerdem die Funktionalitäten Läufe zu erstellen, zu bearbeiten und zu löschen.

## Ziele der Projektarbeit

Ziel der Projektarbeit ist es, die kennengelernten Programmiersprachen HTML, CSS, PHP und JavaScript anzuwenden und durch deren Benutzung die Kenntnisse zu vertiefen und zu festigen. Weiteres Ziel ist es mit den Prinzipien einer modernen Webanwendung zu Arbeiten und dadurch die Abläufe verstehen zu können. Hierbei sind die durch die Aufgabenstellung geforderten Funktionalitäten einzuhalten. Diese umfassen für die Web-Anwendung:

* Umsetzung als Single-Page Interface mit benutzerfreundlicher Oberfläche
* Eine Listenansicht zur Übersicht von Elementen, mit Berücksichtigung auf potenziell sehr groß werdende Anzahl der Elemente
* Eine Detailansicht zu einem beliebigen Element
* Eine Eigenschaft der Elemente soll nicht innerhalb der Datenbank gespeichert, sondern auf Basis vorhandener Daten berechnet werden
* Das Neuanlegen, Bearbeiten und Löschen eines Elements
* Eine Benutzerauthentifizierung
* Dauerhafte Datenbestandsveränderungen nur durch Benutzerbestätigung
* Jedes Element soll von jedem beliebigen authentifizierten Benutzer bearbeitet werden können

Weiterhin sind auf verschiedene Fehlbedienungen und Fehlersituationen zu reagieren, die vor allem durch die Nutzung der Anwendung durch mehrere Benutzer gleichzeitig entstehen können.

Für die mobile Version der Anwendung gilt es lediglich die Listenansicht sowie die Detailansicht zu realisieren.

# Vorgehensweise

## Organisation der Projektarbeit

Um das gegebene Zeitfenster bis zur Abgabe des Projekts halten zu können, wurden zu Beginn des Projekts ein grober Zeitplan erstellt. Die grundsätzliche Organisation wurde definiert, die Aufgaben und erforderlichen Arbeitsschritte benannt und die Verteilung der einzelnen Punkte auf die Gruppe festgehalten.

Die Projektgruppe kam in regelmäßigen Treffen in Planungsrunden zusammen, um sich zunächst genau mit der Aufgabenstellung vertraut zu machen. Die Vorgaben und Anforderungen wurden genau analysiert und festgehalten.

Danach wurde die grundsätzliche Vorgehensweise zur Umsetzung des Projektes festgehalten.

## Aufgaben und Arbeitsschritte

Zunächst wurde ein passendes Thema für die Anwendungsdomäne gewählt. Die Aufgabenstellung beinhaltete dabei schon verschiedene Szenarien, welche nicht erlaubt waren. Nachdem diverse Möglichkeiten gemeinsam besprochen wurden, entschied sich die Gruppe einstimmig für die Entwicklung und Gestaltung einer Sportseite. Daraufhin wurden die einzelnen benötigten Ressourcen definiert, welche für die erfolgreiche Entwicklung des Projekts ausgeschöpft werden müssen.

Erste Arbeitsschritte waren das Erstellen des Seitenlayouts sowie die Datenbankstruktur auf dem MySQL Server. Das Seitenlayout sollte möglichst benutzerfreundlich gestaltet werden und der Domäne entsprechend aussehen. Die Datenbanktabellen müssen alle nötigen Informationen beinhalten, die für den Betrieb der Seite von Bedeutung sind.

Im nächsten Schritt wurde die Entwicklung des Webservice vorgenommen. Dieses Modul stellt die grundsätzlichen Funktionen der gesamten Anwendung zur Verfügung, in das auch spätere Elemente der Seite greifen. Zur Programmierung des Webservice wurde die Programmiersprachen PHP und JavaScript benutzt.

In einer weiteren Aufgabe ging es um die Erstellung des Webclients. Hierzu wurden zunächst mithilfe von HTML und CSS Seitenelemente, Vorlagen und Layout Einstellungen definiert.

Die letzte Aufgabe des Projektes stellte die Implementierung der vollständig fertigen Funktionen der Desktopversion in die entsprechenden Module des Sencha Touch Mobile Client dar.

## Verteilung der Aufgaben auf die Gruppenmitglieder

Am Anfang der Projektplanung wurde vorgesehen, dass nach abgeschlossener Planungsphase gemeinsam mit dem Entwickeln der nötigen Elemente für die Anwendung begonnen wird und das gemeinsame Programmieren auch grundsätzlicher Bestandteil der Treffen wird. Diese Planung wurde im tatsächlichen Verlauf des Projektes zum größten Teil verworfen, da aufgrund von der zeitlichen Knappheit und der gleichzeitig anspruchsvollen Forderungen an das Projekt ein gemeinsames Entwickeln so nicht stattfinden konnte. Stattdessen wurde viel mehr Wert auf das einzelne Entwickeln mit Hilfe des webbasierten Softwareentwicklungs-Hostingdienst GitHub gelegt. Dadurch konnten einfache Entwicklungsaufgaben verteilt und schneller abgearbeitet werden damit bei den Projekttreffen effektiver auf schwierigere Elemente eingegangen werden konnte und diese gemeinsam bearbeitet werden. Diese Vorgehensweise wurde vor allem während der Entwicklung der Anwendung angewendet.

# Anforderungsanalyse

## Funktionale Anforderungen

Zur Übersicht der Funktionalen Anforderungen wurde ein Use-Case Diagramm erstellt:

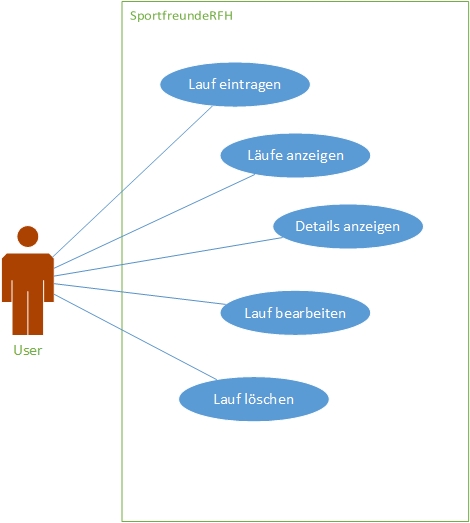


Abbildung Use-Case Diagramm der SportfreundeRFH Seite

Die Use-Cases stellen die einzelnen Anwendungsfälle der Desktopversion dar. Bei der mobilen Version beschränken sich die Anwendungsfälle auf „Läufe anzeigen“ und „Details anzeigen“.

In jedem Anwendungsfall wird ein *Request* vom Client an den Webservice gestellt und dieser liefert als Antwort ein Objekt im JSON-Format zurück.

Dabei überprüft der Webservice die Art der gestellten Anfrage(GET, POST, PUT, DELETE) und reagiert dementsprechend mit der richtigen Methode. Sollte es zu einem Fehler kommen, liefert der Webservice den entsprechenden http-Code als Antwort zurück.

Im Folgenden wird genauer auf die einzelnen Anwendungsfälle eingegangen:

* **Läufe anzeigen**

In diesem Use-Case wird ein GET-Request an den Webservice gesendet. Dieser liefert als Antwort die komplette Liste aller Tracks in der Datenbank zurück. Die Liste kann dann an entsprechender Stelle im Webclient dargestellt werden.

* **Details anzeigen**

Die Details werden ebenfalls über einen GET-Request vom Webservice abgerufen, jedoch wird bei der Anfrage an den Webservice eine konkrete ID mitgegeben, um genau den geforderten Track als Antwort zu bekommen und anzeigen zu können.

* **Lauf eintragen**

Um einen Lauf eintragen zu können, wird eine POST-Anfrage an den Webservice gestellt. Hierzu werden die vom Benutzer eingetragenen Attributwerte in den Eingabefeldern ausgelesen an den Webservice übertragen. Dieser baut dann die Datenbankverbindung auf und speichert die übergebenen Werte entsprechend ab.

* **Lauf bearbeiten**

Zum Bearbeiten eines Eintrags wird eine PUT-Anfrage gestellt. Dabei können in der Datenbank einzelne Werte verändert werden, jedoch nur, wenn in der Zwischenzeit nicht bereits eine Veränderung stattgefunden hat. Hier wurde eine optimistische Nebenläufigkeitskontrolle über Versionierung realisiert.

* **Lauf löschen**

Um einen Lauf zu löschen, wird ein DELETE-Request an den Webservice gesendet. Hierbei wird auch wieder anhand der ID ein bestimmter Track identifiziert und gelöscht.

Für die Use-Cases der mobilen Version werden dieselben Anfragen beim Anzeigen der kompletten Liste und der Detailansicht wie bei der Desktopversion an den Webservice gestellt.

## Nichtfunktionale Anforderungen

Nichtfunktionale Anforderungen stellen typischerweise Anforderungen an die Oberfläche, Datenbanken sowie Performance, Verfügbarkeit und Portierbarkeit dar.

Dazu wurde während der Erstellung darauf geachtet, ein möglichst benutzerfreundliches Layout zu erstellen. Die Oberfläche ist klar strukturiert und übersichtlich. Bedienelemente sind eindeutig platziert und benannt.

Die Anzahl der Datenbankzugriffe sollte möglichst klein gehalten werden, um eine hohe Performance zur gewährleisten.

Durch den modularen Aufbau aller Komponenten im System sind diese auch durchaus portierbar auf andere Anwendungen.

# Technische Umsetzung

## Eingesetzte Anwendungen

Die zur Umsetzung des Projektes zulässigen und benutzten Anwendungen wurden bereits während der Vorlesung kennengelernt. Diese sind nachfolgend aufgelistet:

* **GitHub**

Webbasierter Hosting-Dienst für Softwareentwicklung. Vereinfacht das gemeinsame Arbeiten an den Quellcode-Dateien durch automatisches Dokumentieren von Änderungen und bietet die Möglichkeit durch Datensynchronisation für alle beteiligten Entwickler immer auf demselben Stand zu sein

* **Notepad++**

Einfacher, freier Texteditor, welcher das Bearbeiten von Quelltext besonders unterstützt. Bietet hilfreiche Funktionen für eine Vielzahl von Programmiersprachen durch Syntaxunterstützung und vereinfacht die Leserlichkeit durch farbliche Hervorhebungen

* **Fiddler**

Debugg-Tool für http anfragen, Filtert den gesamten http Verkehr als Proxy zwischen Client und Server zur Analyse entstehender Fehler und speziell für das Projekt zur Kontrolle, ob bestimmte Anfragen korrekt durchgeführt werden

* **XAMPP**

Umfangreiches Softwarepaket zum einfachen konfigurieren einer Webserverumgebung, enthält unteranderem Apache Webserver, MySQL Datenbank und die Skriptsprache PHP

* **Browser**

Programm zum Darstellen von Websites. Im Projekt kam vor allem Google Chrome zum Einsatz, da dieser nützliche Werkzeuge zum Debuggen von Websites besitzt

## Aufgaben der Anforderungen

## Abläufe innerhalb der Software

Das folgende Zustandsdiagramm zeigt eine Übersicht der Desktopversion des Webclients:

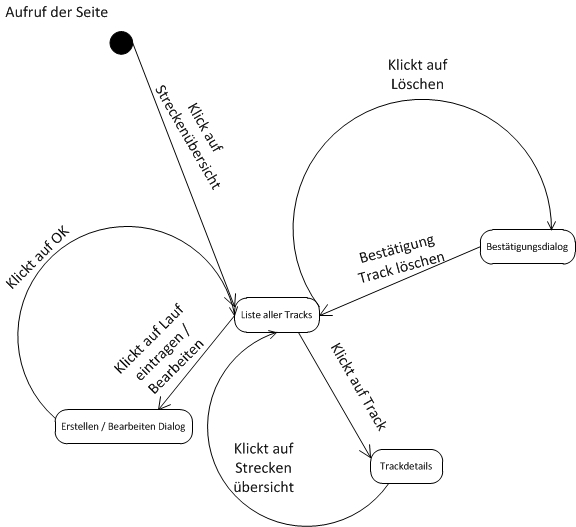


Abbildung Zustandsdiagramm

Die innerhalb der Webanwendung ablaufenden Sequenzen werden in den folgenden Beiden Diagrammen beispielhaft dargestellt:

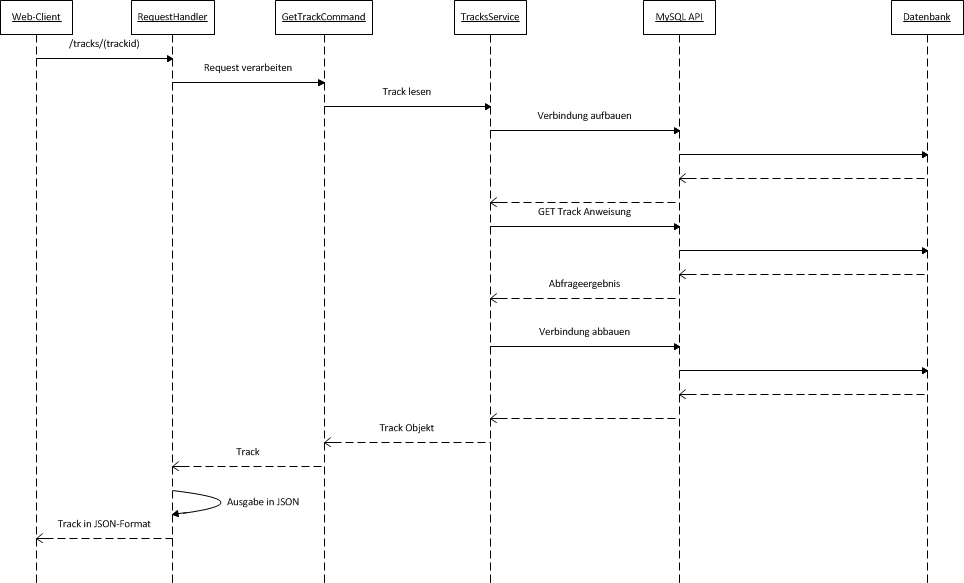


Abbildung Sequenzdiagramm zur Darstellung der Kommunikation

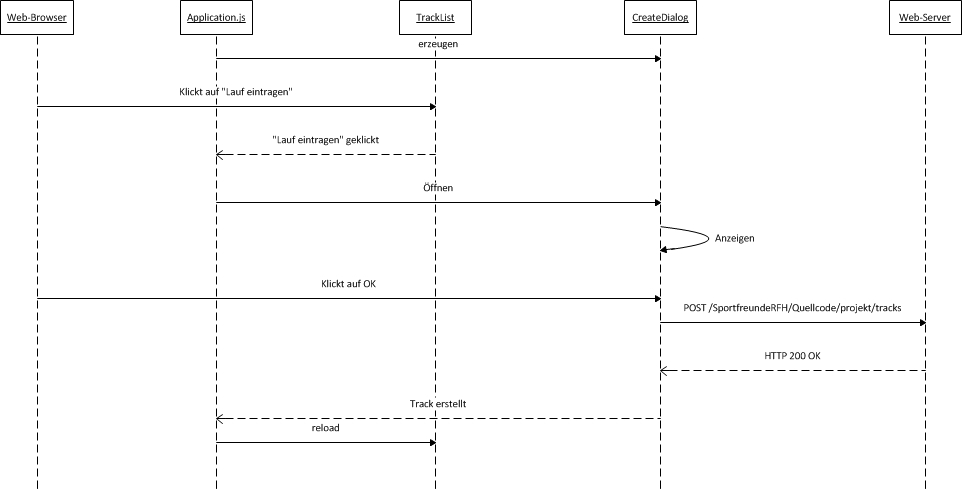


Abbildung Sequenzdiagramm zur Darstellung des Datenfluss

## Strukturelemente der Software

Es folgt eine Veranschaulichung der statischen Struktur des Webservice:

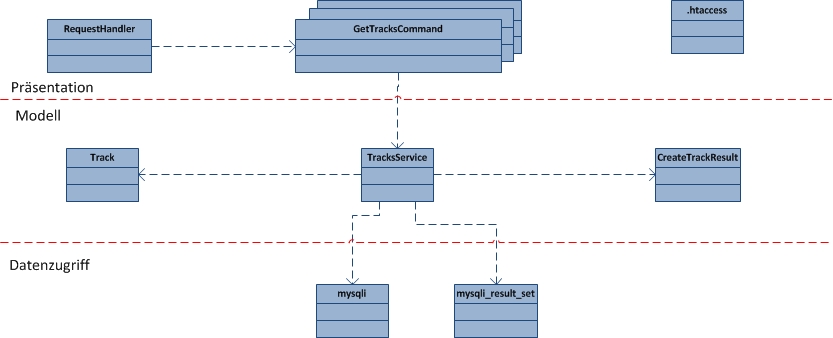


Abbildung Struktur des Webservice

Sowie eine Abbildung um die Struktur des Webclient darzustellen:

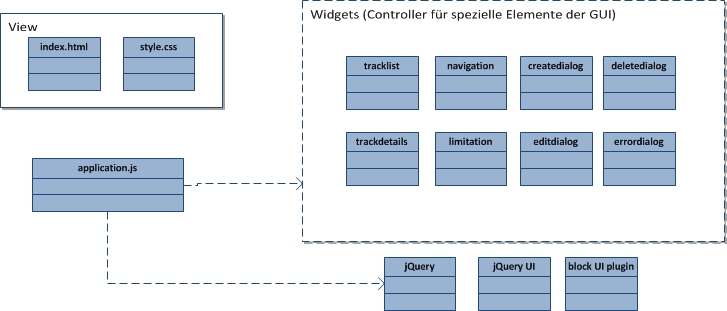


Abbildung Struktur des Webclient

## Datenmodell-Entwurf

Die nachfolgende Abbildung zeigt das Datenmodell der Datenbank zur Webanwendung:

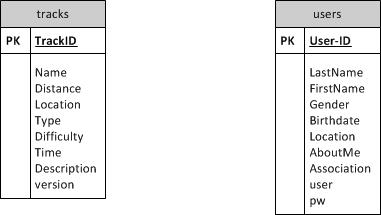


Abbildung Datenmodell

Hierbei ist zu beachten, dass im aktuellen Stand der Entwicklung die Tabellen untereinander noch keinerlei Beziehungen haben. Die Tabelle users dient aktuell lediglich zur Authentifizierung eines Benutzers mittels seines gewählten Usernames und Passwort. Zukünftig ist es vorgesehen, dass jeder erstellte Track genau dem erstellenden User zugeordnet werden kann.

# Eingereichte Ergebnisse

**Webclient:**

**Hauptverzeichnis:** */SportfreundeRFH/Quellcode/projekt*

Enthält alle benötigten Ordner und Daten für die Anwendung

**Inhalt:**  *.htaccess*

Regelt http Umleitungen für den Webservice

*index.php*

Hauptseite der Anwendung, beinhaltet HTML/PHP Code zur Darstellung der Inhalte

*login.php*

PHP Seite zur Authentifizierung der Benutzer

**Unterordner:** */css*

Daten, die für die Darstellungder Seite verantwortlich sind

**Inhalt:** *jquery-ui-1.10.4.css*

jQuery eigene Datei zur Darstellung von jQuery UI Elementen

*style.css*

Datei für die Darstellung der HTML Elemente des Webclient

**Unterordner:** */img*

Beinhaltet alle Bild-Dateien für die Anwendung

*bg.jpg*

Hintergrundbild

*header2.png*

Schriftzug in der Kopfzeile der Seite

**Unterordner:**  */js*

Beinhaltet alle Javascript Daten

*application.js*

Instanziiert die weiteren Javascript Klassen, kontrolliert Widgets und löst definierte Trigger aus

*jquery.blockUI.js*

jQuery Bibliothek mit Methoden zur Verhinderung von ungewünschten User Aktivitäten

*jquery-1.11.2.min.js*

jQuery Hauptbibliothek als minimierte Version

*jquery-ui-1.10.4.js*

jQuery Bibliothek zur Gestaltung und Funktionalität der Benutzeroberfläche

*track.createdialog.js*

Widget, zum Anzeigen des Erstellendialogs

*track.deletedialog.js*

Widget, zur Abfrage der Löschbestätigung

*track.editdialog.js*

Widget, zum Anzeigen des Bearbeitendialogs

*track.errordialog.js*

Widget, welches Fehlermeldungen anzeigt

*track.limitation.js*

Widget, zur Steuerung des Listenpaging

*track.navigation.js*

Widget, für die Seitennavigation

*track.trackdetails.js*

Widget, zur Detailansicht eines Listeneintrags

*track.tracklist.js*

Widget, zur Erstellung der Liste

**Unterordner:**  */webservice*

Beinhaltet die PHP Daten zur Bereitstellung der Controller Funktionalität

*CreateTrackCommand.php*

Regelt die Erstellung eines Tracks

*CreateTrackResult.php*

Regelt die Übergabe von Fehlermeldungen eines Track Objekts

*DeleteTrackCommand.php*

Regelt das Löschen eines Tracks

*GetTrackCommand.php*

Sorgt für die Anzeige eines einzelnen Tracks

*GetTracksCommand.php*

Sorgt für die Anzeige der gesamten Track Liste

*RequestHandler.php*

Regelt die http Anfragen an den Service

*Track.php*

Klasse die Attribute eines Track Objektes definiert

*TracksService.php*

Steuert die Funktionalität des Service

*UpdateTrackCommand.php*

Regelt das Bearbeiten von Tracks

**Mobile Client:**

**Hauptverzeichnis:** */SportfreundeRFH/Quellcode/projekt/mobile*

Enthält die relevanten Daten von Sencha Touch

**Inhalt:**  *app.js*

Hauptdatei von Sencha Touch

*index.html*

Datei zur Darstellung der Inhalte

**Unterordner:** */.sencha*

Enthält Daten für Sencha Touch

**Unterordner:** */app*

Enthält die relevanten Javascript Daten

**Inhalte:**  */controller/AppController.js*

Steuert den Mobile Client

*/model/Track.js*

Modeldatei für die Trackliste

*/store/Tracks.js*

Modeldatei für einen einzelnen Track

*/view/Main.js*

Hauptdatei zur Anzeige der Liste

*/view/TrackForm.js*

Datei zur Anzeige der Track Details

*/view/TrackList.js*

Datei zur Anzeige der Track Liste

**Unterordner:**  */resources*

Enthält Daten für Sencha Touch

**Unterordner:**  */touch*

Enthält Daten für Sencha Touch

**Datenbankdatei:** */SportfreundeRFH/sportfreunde.sql*

Enthält die Datenbankstruktur und Testdatensätze

# Installationsanleitung

**Vorraussetzungen:**

* Installierte Version des aktuellen XAMPP Paket

**Installation:**

1. Bitte Starten Sie über das XAMPP Control Panel die Module „Apache“ sowie „MySQL“
2. Bitte Entpacken Sie das .zip-Archiv „SportfreundeRFH.zip“
3. Kopieren Sie den Ordner „SportfreundeRFH“ in den XAMPP Ordner htdocs
   * Standard unter Windows C:\xampp\htdocs
4. Rufen Sie über Ihren Browser *localhost/phpmyadmin* auf
   * Zugangsdaten für den mySQL Server sind standardmäßig „root“ und kein Passwort
5. Legen Sie bitte eine neue Datenbank mit dem Namen „sportfreunde“ an
6. Importieren Sie in diese Datenbank die Daten aus der *sportfreunde.sql* Datei
7. Sie erreichen die Startseite der Anwendung nach Eingabe folgender URL in den Browser: <http://localhost/SportfreundeRFH/Quellcode/projekt/index.php>
8. Die Login-Daten lauten:

* **Username:** dozent
* **Passwort:** 123

**Hinweis:**

Sollten Sie für Ihre MySQL Datenbank veränderte Login Daten haben, ändern Sie diese bitte in der Datei „TracksService.php“ (Zeile 13) im Ordner „webservice“.

# Fazit

Kritische Würdigung, Stärken und Schwächen sowie Grenzen der Projektarbeit, Ausblick

Durch die Teilnahme an den Vorlesungen und das durchführen dieses Webprojektes erfuhren alle an dem Projekt beteiligten Mitglieder einen sehr intensiven Lerneffekt im Bereich des Webdevelopments. Hierbei wurden in einer sehr verdichteten Form die nötigen Grundkenntnisse der angewendeten Programmiersprachen und Prinzipien einer modernen Webanwendung vermittelt und konnten während der Entwicklungsphase gefestigt und ausgebaut werden.

Wie schon in den Vorlesungen deutlich wurde und auch im Projekt genauso umgesetzt, ist die Modularisierung der wichtigen Funktionen. Dadurch entsteht eine Art Baukasten-Prinzip, welches es gerade für Anfänger leicht verständlich macht, wie die einzelnen Komponenten miteinander arbeiten. Hieraus entsteht im Endeffekt aber auch eine Arbeitserleichterung für das gesamte Konstrukt, da spätere Weiterentwicklungen einfach implementiert oder ganze Module getauscht werden können.

Die Performance des Paging der Liste leidet bei größer werdender Anzahl Elementen. Da hierbei die Liste auf jeder angezeigten Seite aktualisiert wird, damit der User zwischenzeitliche Änderungen direkt sieht, entstehen viele Requests an den Server. An dieser Stelle könnte die Performance gesteigert werden.

In Bezug auf das Layout der Seite wurde innerhalb des Projektes weniger Zeit eingeplant, als für das Umsetzen der geforderten Funktionalitäten. Dieses ist sehr einfach und weitestgehend Textgestützt umgesetzt. Dadurch entspricht es nicht dem modernen Layout einer heutigen Website und ist durchaus verbesserungswürdig.

In der Planungsphase der Projektarbeit gab es vielerlei gute Ideen die Funktionalität der Website zur erhöhen. So sollte es unter anderem für jeden Benutzer ein eigenes Profil geben, in dem der Läufer sich präsentieren kann. Des Weiteren gab es die Idee, jedem User seinen eigenen Bereich für Trainingsläufe einzurichten, sowie einen Wettkampfbereich für spezielle Läufe (z.B. Großveranstaltungen wie den Köln Marathon) in dem sich die Benutzer gegenseitig vergleichen und in einer Bestenliste geführt werden können. Leider wurde im Laufe des Projektes schnell festgestellt, dass all diese Ideen nicht implementiert werden können, da aufgrund des eng gesteckten zeitlichen Rahmen der Fokus auf die geforderten Ziele, zum erfolgreichen Beenden des Projektes, im Vordergrund stehen musste.

# Quellenverzeichnis

[www.w3schools.com](http://www.w3schools.com)

[www.php.net](http://www.php.net)

[www.stackoverflow.com](http://www.stackoverflow.com)