



**HSR**

HOCHSCHULE FÜR TECHNIK  
RAPPERSWIL

FHO Fachhochschule Ostschweiz



# Android Applikation RadioTour

Studienarbeit

Abteilung Informatik  
Hochschule für Technik Rapperswil

Frühjahrssemester 2012  
30. Mai 2012

*Autoren:* Florian Bentele & Daniel Stucki  
*Betreuer:* Prof. Dr. Peter Heinzmann  
*Projektpartner:* cnlab AG, Rapperswil-Jona



Die Android Tablet Applikation *RadioTour* ersetzt die bisherige webbasierte Anwendung zur Erfassung der Rennsituation an der Tour de Suisse. Der *RadioTour Speaker* verfolgt in einem Begleitfahrzeug die Radfahrer und zeichnet Veränderungen und Rückstände auf. Die Informationen erhält er direkt von Helfern aus Begleitfahrzeugen.

Zu Beginn einer Etappe wird die Fahrerliste in die Applikation importiert, dabei ist auch die aktuelle Klassifizierung (Rangliste) vom Vortag. Änderungen können während einer Etappe aufgezeichnet und abgespeichert werden. Die native Bedienung mit Finger ermöglicht es, auf dem Tablet Radfahrer zu gruppieren und deren zeitliche Abstände zu erfassen. Aus diesen Angaben wird ein virtuelles Klassement erstellt. Diese live aufgezeichneten Informationen werden über das Mobilfunknetz auf die Server der TourLive Webseite übertragen.

Da die Applikation genau für den RadioTour Speaker angepasst ist, wird sie nicht für die Öffentlichkeit zur Verfügung stehen. Deshalb wird bei der Entwicklung nicht auf die Portierbarkeit für verschiedene Plattformen bzw. Gerätetypen geachtet, sondern vielmehr für einen konkreten Typ optimiert. Das für den RadioTour Speaker vorgesehene Endprodukt beinhaltet ein Tablet mit der RadioTour Applikation und einer vorkonfigurierten Verbindung zum Server.



---

# Aufgabenstellung

Studiengang: Informatik (I)  
Semester: FS 2012 (20.02.2012-16.09.2012)  
Institut: ITA: Internet-Technologien und Anwendungen  
Gruppe: Florian Bentele, Daniel Stucki  
Verantwortlicher: Dr. Prof. Peter Heinzmann  
Industriepartner: cnlab AG, Lukas Frey

---

## Ausgangslage Studienarbeit „RadioTour“

Bei Radrennen erfasst der so genannte RadioTour-Speaker Informationen zur Rennsituation, welche ihm von Motorradfahrern per Funk geliefert werden. Gegenwärtig legt der RadioTour-Speaker mit Hilfe einer TabletPC Web-Anwendung per Click auf die erhaltenen Fahrernummern die Zusammensetzung der Gruppen fest. Er tippt auch die Zeitabstände zwischen den Gruppen ein. Die RadioTour-Anwendung visualisiert die Fahrergruppen und liefert Detailinformationen zu den Fahrern (z.B. Namen, Team, Virtueller Rang). Veränderungen in den Fahrergruppen können per Drag-and-Drop nachgeführt werden. Die mit der RadioTour-Anwendung erfassten Rennsituationen werden per Mobilfunknetz zu einem Webserver gesendet, wo sie für weitere Anwendungen, z.B. für Live Webinformationen zu Verfügung stehen.

---

## Ziel

Ziel dieser Arbeit ist es, die bestehende Web Applikation Tour Live, in eine native Android Tablet Applikation zu portieren. Die Applikation richtet sich einem sehr spezifischen Umfeld, daher ist die Bedienung diesen Anforderungen anzupassen. Das Endprodukt beinhaltet die funktionierende Applikation mit der Schnittstelle zum Server für die Übertragung der Daten. Die Applikation ermöglicht es dem RadioTour-Speaker alle Angaben gemäss Requirements zu erfassen und bietet die Möglichkeit zur Mehrsprachigkeit.

---

## Teilaufgaben

- Analyse der existierenden RadioTour- und TourLive-Anwendungen
  - Tour de Suisse Dokumente
  - Alte RadioTour-Anwendung  
<http://gps.cnlab.ch/tablet/>

user: ba\_tourlive  
password: access4tl

- TourLive-System (GPS Positions- und Bilderfassungssysteme, Webanwendung)
- Kommunikation RadioTour – TourLive-Webanwendung
- Vergleich mit Systemen anderer Radrennen
- Festlegung der Funktionalität der neuen Tablet-Anwendung (Requirements Engineering)
  - Studium der Geschäftsprozesse (Renninformationen, Rennverlauf)
  - Austesten von Teilfunktionen (Android-Tablet Programmierung, GPS-Positionserfassung, Usability Experimente)
  - Erweiterte Funktionen (z.B. Erfassung der Streckenkilometer, Integration von Marschtabellen)
  - Auswahl der Hardware-Plattform
  - Spezifikation
- Design Benutzerschnittstelle Kommunikation mit TourLive Aufnahmesystemen (Weitergabe der Daten an Datenserver) RadioTour-Anwendung
- Realisierung Prototypen Anpassungen Friendly User Test Version (Beta-Release) Feldtest an einem Radrennen Übergabe an cnlab
- Dokumentation Gemäss Anforderungen Industriepartner (das System soll vom Industriepartner betrieben und erweitert werden können) Bericht gemäss HS-R/Heinzmann Richtlinien

---

## Abgrenzung

Das Produkt wird spezifisch auf ein Gerät ausgerichtet und nicht plattformübergreifend entwickelt. Die Verbindung zum Server wird in der Arbeit definiert jedoch werden keine serverseitigen Entwicklungen erarbeitet.

Die Mehrsprachigkeit wird nach Android Standards implementiert <sup>1</sup>. Eine Übersetzung ist jedoch nicht Teil der Arbeit.

---

## Erklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selber und ohne fremde Hilfe durchgeführt habe, ausser derjenigen, welche explizit in der Aufgabenstellung erwähnt ist oder mit dem Betreuer schriftlich vereinbart wurde, dass ich sämtliche verwendeten Quellen erwähnt und gemäss gängigen wissenschaftlichen Zitierregeln korrekt angegeben habe.

Rapperswil, 29. Mai 2012

Florian Bentele

Daniel Stucki

---

1. <http://developer.android.com/guide/topics/resources/localization.html#creating-alternatives>

---

# Management Summary

---

## Ausgangslage

An der Tour de Suisse fahren ca. 200 Radrennfahrer in Tagesetappen durch die ganze Schweiz. Dabei werden Sie von diversen Motorfahrzeugen begleitet. Im Feld fährt ebenfalls der RadioTour Speaker mit. Seine Funktion besteht darin, Live Informationen des Rennens zu erfassen und an den Server der cnlab AG weiterzuleiten. Die Übertragung der Daten vom Gerät zum Server geschieht über das Mobilfunknetz 3G.

### Live Informationen

Während dem Rennen werden aus verschiedenen Quellen Informationen gesammelt. Zum einen sind dies Veränderungen im Rennfeld, zum anderen sind dies Wertungen, die die Fahrer erreichen können, so z.B. einen Bergsprint. Diese Daten werden vom RadioTour Speaker manuell erfasst.

Wenn sich ein Rennfahrer vom Feld ablöst und einen Vorsprung erarbeitet so wird dieser von einem Motorradfahrer verfolgt. Diese Änderung wird dann sofort per Funk an den RadioTour Speaker übermittelt.

### Äussere Bedingungen

Bei Live Sport Events wie der Tour de Suisse ist die Erfassung von Echtzeitdaten, aus technischer Sicht, eine Herausforderung. Die Bedingungen werden erschwert zum einen durch das Alpine Gebirge wo die Mobilfunkverbindungen und GPS Informationen nicht immer gewährleistet sind, zum anderen durch die ständigen Vibrationen der Fahrzeuge.

Die Unterbrüchen der Verbindung werden überbrückt, indem die Änderungen gesammelt und periodisch an den Server gesendet. Alle Änderungen werden als Paket in eine Warteschlange eingetragen, ist ein Übertragen zum Zeitpunkt nicht möglich wird es später wieder versucht.

---

## Vorgehensweise

In dieser Studienarbeit kommt das Vorgehensmodell zur Softwareentwicklung von Rational Unified Prozess (RUP) zur Anwendung. Das Projekt wird in die folgenden vier Phasen aufgeteilt:

- Inception
- Elaboration

- Construction
- Transition

In jeder dieser Phase werden die Arbeitsschritte nach RUP durchgeführt, je nach dem in mehreren Iterationen wie es bei dieser Arbeit in der Phase *Construction* vorkommt.<sup>2</sup>

Die Erfassung der Anforderungen, der Entscheid zur Entwicklung auf einem Android Gerät sowie die Evaluation eines geeigneten Tablets bilden zusammen die Startphase des Projekts. Die Kriterien auf der die Entscheidung gestützt sind, wurden in einer Sitzung zusammen mit Herrn Dr. Prof. Peter Heinzmann, dem Betreuer der Arbeit diskutiert und genehmigt.

Im weiteren Verlauf der Arbeit werden die Anforderungen und die UseCases definiert. Daraus entsteht dann die Domainlogik und parallel dazu einen ersten Prototypen des UserInterfaces. Insbesondere die Benutzerschnittstelle entsteht in mehreren Iterationen, da die Bedienung massgebend am Erfolg des Produktes beteiligt ist und erst bei der Anwendung ersichtlich wird ob die Bedienung optimal ist.

---

## Ergebnisse

Die RadioTour Android Applikation beinhaltet die festgelegten Anforderungen. Die Fahrerlisten und die offiziellen Zeitmessungen können via USB oder über das Internet importiert werden. Die Gruppen lassen sich dynamisch verändern und die Rennsituation wird an den Server übermittelt. Ein Testlauf mit dem ersten Prototyp hat klar aufgezeigt, dass die Anwendung eine Verbesserung in der Bedienung bringt. Dabei entstehen keine Einbussen in der Funktionalität. Für den Einsatz an der Tour de Suisse ist die Applikation bereit.

---

## Ausblick

Für den erfolgreichen Einsatz an der Tour de Suisse ist ein Testlauf, insbesondere um die Serververbindung zu testen, notwendig.

Nach der Tour de Suisse müssen die Eindrücke und das Feedback des RadioTour Speakers aufgenommen werden, damit die Applikation weiter verbessert werden kann.

---

2. Frei nach [http://de.wikipedia.org/wiki/Rational\\_Unified\\_Proces](http://de.wikipedia.org/wiki/Rational_Unified_Proces)



---

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>11</b>
1.1	BigPicture . . . . .	11
1.2	Evaluation und Kaufempfehlung . . . . .	12
<b>2</b>	<b>Analyse</b>	<b>15</b>
2.1	Requirements . . . . .	15
2.1.1	Funktionale Anforderungen . . . . .	15
2.1.2	Nicht funktionale Anforderungen . . . . .	16
2.2	Struktur der Applikation . . . . .	16
2.3	Technologien . . . . .	16
2.3.1	Android . . . . .	16
2.3.2	Externe Libraries . . . . .	17
2.3.3	Entwicklungsumgebung . . . . .	17
2.3.4	Android Version . . . . .	17
2.4	Mitbewerberanalyse . . . . .	18
<b>3</b>	<b>Architektur</b>	<b>19</b>
3.1	Klassendiagramm . . . . .	19
3.2	Sequenz Diagramm . . . . .	19
<b>4</b>	<b>Realisierung</b>	<b>23</b>
<b>5</b>	<b>Testing</b>	<b>25</b>
5.1	Java Unit Tests & Android Testing Framework . . . . .	25
5.2	Feldtest . . . . .	25
<b>6</b>	<b>Ergebnisse und Schlussfolgerungen</b>	<b>29</b>
<b>7</b>	<b>Verzeichnisse und Referenzen</b>	<b>31</b>
7.1	Literaturverzeichnis . . . . .	31
7.2	Tabellenverzeichnis . . . . .	31

7.3	Abbildungsverzeichnis . . . . .	31
<b>Glossar</b>		<b>35</b>
<b>8</b>	<b>Anhang</b>	<b>39</b>
8.1	BigPicture . . . . .	39
8.2	Projektmanagement . . . . .	39
8.2.1	Zeitplanung . . . . .	39
8.2.2	Code Base und Issue Tracking . . . . .	41
8.3	Persönliche Berichte . . . . .	41
8.3.1	Daniel Stucki . . . . .	41
8.3.2	Florian Bentele . . . . .	41
8.4	Software Dokumente . . . . .	42
8.5	Kriterienkatalog . . . . .	42
8.6	Kaufempfehlung . . . . .	42
8.7	UseCases der bisherigen Applikation . . . . .	42
8.8	Usability Test . . . . .	46
8.8.1	Auswertung und Feedback . . . . .	47



Die schematische Darstellung zeigt im wesentlichen die drei Hauptaktoren auf. Zum einen sind dies die Motorradfahrer, welche die Radrennfahrer begleiten und Veränderungen in Echtzeit per Funk übermitteln. Diese Informationen kommen in kurzen Abständen und müssen sofort erfasst werden können. Im UserInterface verwenden wir dafür eine Lösung bei der mehrere Radrennfahrer gleichzeitig eingetragen werden können.

Eine weitere Rolle spielt der *RadioTour Speaker* mit dem Android Tablet. Er fasst die Informationen zusammen und wertet diese bereits auf dem Gerät aus. Im Tablet werden auch Daten wie z.B. die Durchschnittsgeschwindigkeit und die aktuelle Rennzeit angezeigt.

Der dritte Akteur bildet der Server der *cnlab AG*, welcher direkt mit der Applikation kommuniziert. Ausgetauscht werden die Veränderungen im Feld sowie Rückstände von der Spitze. Weiter können Ereignisse wie z.B. eine Verletzung oder ein defektes Fahrrad aufgezeichnet werden. Die Daten werden dann weiter auf der Webseite der *TourLive* aufbereitet und publiziert. Nicht nur für die beteiligten im Team sondern auch für Fans sind diese Angaben von grossem Interesse, da die Daten vor den offiziellen Zeitmessungen bereits einen Einblick in das Schlussklassement geben. Das BigPicture ist in voller grösse ebenfalls im Anhang 8.1.

---

## 1.2 Evaluation und Kaufempfehlung

Die Evaluation der Zielplattform war ein wichtiger Faktor für die weitere Entwicklung der Arbeit. Aus diesem Grund stand dies ganz zu Beginn der Arbeit an. Zur Auswahl standen die beiden marktführenden Betriebssysteme Android (Google) und iOS (Apple). Als Grundlage für die Evaluation dienten die folgenden Kriterien:

- Vorkenntnisse der Programmiersprachen Java bzw. Objective-C
- Möglichkeiten zum UserInterface Design
- Programmierumgebung, Integrated Development Environment (IDE)
- mögliche Vertriebskanäle der Applikation
- Nutzbarkeit von externen Geräten und Schnittstellen
- Vielfalt von Informationsquellen im Internet

Die Kriterien werden in einer Nutzwertanalyse gewichtet und bewertet. Insbesondere die Vorkenntnisse in Java sind ausschlaggebend für den Entscheid, die Applikation für die Androidplattform zu entwickeln. Dieser Entscheid ist in Absprache mit Herrn Heinzmann getroffen worden. Die gesamte Liste der Kriterien mit der jeweiligen Gewichtung sowie eine ausführliche Erläuterung befinden sich im Anhang ??.

Für die Auswahl eines geeigneten Tablets wird im nächsten Schritt ein Kriterienkatalog definiert mit zwingenden und optionalen Kriterien für das Gerät. Die zwingenden Kriterien beinhalten:

- Android Betriebssystem, gemäss Evaluation
- USB Anschluss für den Import der Fahrerliste am Renntag, optional auch mit Adapter möglich

- Mobilfunknetz 3G für die Kommunikation mit dem Server
- GPS für die Lokalisierung
- Stromversorgung durch 12V (Auto) Adapter möglich

Zu den optionalen Kriterien gehören die Akkulaufzeit, falls die Stromversorgung unterbrochen wird sowie ein grosszügiger Bildschirm für die Bedienung mit dem Finger oder mithilfe eines Stiftes.

Als Sieger und somit auch als Kaufempfehlung an die *cnlab AG* geht das Lenovo ThinkPad Tablet. Dieses Gerät erfüllt alle Kriterien und überzeugt in der Vielfalt der Anschlüsse. Die Kaufempfehlung mit weiteren Erläuterungen ist ebenfalls im Anhang ?? zu finden.

Im Verlauf der Arbeit ist ein defekt an der Micro USB Buchse entstanden. Dieser Anschluss wird für die Entwicklung auf dem Gerät dringend benötigt. Für die weitere Entwicklung ist ein Ersatzgerät angeschafft worden. Dabei handelt es sich um das Galaxy Nexus Tab 10.1<sup>1</sup>

---

1. <http://www.samsung.com/ch/consumer/mobile-phone/tablets/tablets/GT-P7500UWDITV> aufgerufen am 23.05.2012



Die Analyse untersucht die bestehende Web Applikation und fasst die Anforderungen zusammen. Weiter wird die Struktur der Applikation schematisch dargestellt. Es werden die verwendeten Technologien sowie einen kurzen Exkurs zu anderen Lösungen angesprochen.

---

## 2.1 Requirements

Die Anforderungen an die *RadioTour* Applikation ergeben sich aus den Features der bisherigen Web Applikation und den Verbesserungsvorschlägen des *RadioTour* Speakers. Im Anhang 8.7 sind sämtliche UseCases der bisherigen Applikation aufgeführt. Im folgenden Abschnitt werden die Funktionen, welche in diesem Projekt implementiert sind, aufgeführt.

### 2.1.1 Funktionale Anforderungen

Die funktionalen Anforderungen sind in drei Prioritätsstufen eingeteilt:

#### **zwingenden Anforderungen (must)**

- Fahrerliste, Etappen und Marschtabellen importieren
- Fahrer ansehen, sortieren und bearbeiten
- Gruppen bilden und Rückstand angeben
- Gruppen auflösen
- Events für Fahrer erfassen (Sturz, Arzt, Aufgabe)
- Rennsituation an den Server übermitteln
- Spezialklassemente und Wertungen erstellen und Fahrer zuweisen
- Maillots erfassen und bearbeiten
- Persistierung der Daten auf dem Gerät

#### **optionalen Anforderungen (can)**

- aktuelle Rennkilometer und Rennzeit anzeigen
- Stoppuhr
- aktuelle Position durch GPS bestimmen

- Events für Fahrer an den Server übermitteln

#### **wünschenswerten Anforderungen (nice to have)**

- aktuelle Position des *RadioTour Speakers* in der Marschtabelle anzeigen.
- Splashscreen beim Start der Applikation
- 

#### **2.1.2 Nicht funktionale Anforderungen**

Im Umfeld eines Live Sport Events spielen neben den funktionalen Anforderungen an ein Software Produkt auch nicht funktionale Anforderungen eine wichtige Rolle. So muss z.B. das UserInterface Fehler bei der Eingabe tolerieren bzw. korrigierbar machen. Die Bedienung muss flüssig verlaufen und zu jedem Zeitpunkt muss der Status der Applikation sichtbar sein. Die für *RadioTour* relevanten nicht funktionalen Anforderungen sind im folgenden aufgelistet.

- Mobilfunkverbindung ist nicht immer Gewährleistet, keine Daten dürfen dadurch verloren gehen
- Informationen müssen auch bei direktem Sonnenlicht gut lesbar sein
- Angenehme, flüssige und selbsterklärende Bedienung der Applikation

Weitere nicht funktionale Anforderungen beziehen sich auch auf die Hardware an sich und sind somit im Kapitel 1.2 bereits aufgeführt.

---

## **2.2 Struktur der Applikation**

Die Applikation hat im Grunde zwei Status, einerseits werden vor dem Rennen die Fahrerliste und die Marschtabelle importiert andererseits wird die Rennsituation während dem Rennen erfasst und Änderungen festgehalten. Diese beiden Status können aber nicht absolut voneinander getrennt werden, da während dem Rennen Änderungen denkbar sind. Während dem Rennen müssen gewisse Daten immer angezeigt werden. Diese Live Informationen werden deshalb als eigene Ebene abgebildet. Aus den Anforderungen und den Kriterien entsteht die Baumartige Struktur, wie sie in der Abbildung zu sehen ist.

Der Stamm stellt die Applikation dar und die Äste zeigen die Aufteilung der Funktionen. Die Rennzeit sowie die aktuelle Rennposition sind in einem immer sichtbaren Bereich platziert.

Die untere Ebene beinhaltet die Kernelemente der Applikation und diese werden in Views seitenweise dargestellt. Durch eine Navigation lässt sich zwischen den Views wechseln ohne, dass dabei die Live Informationen ausgeblendet werden.

---

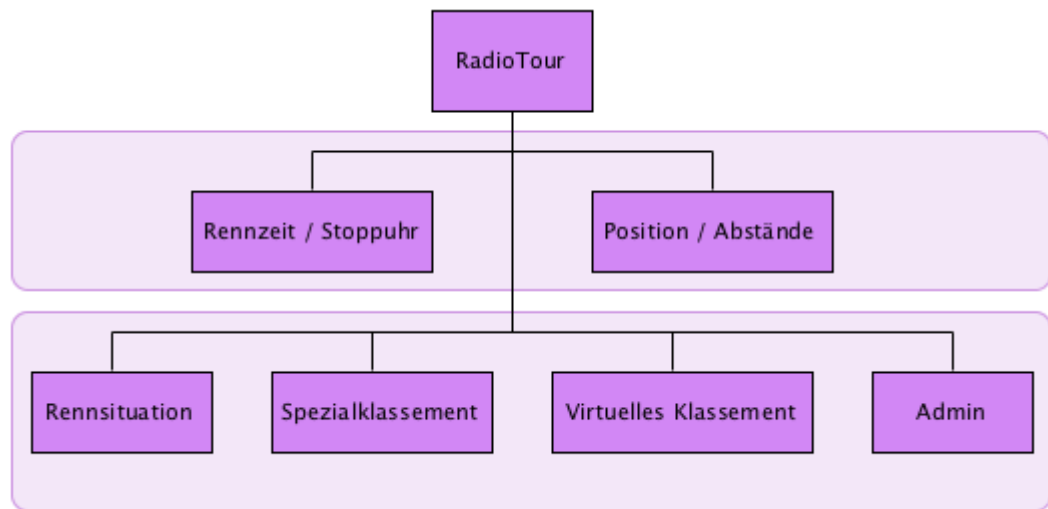
## **2.3 Technologien**

### **2.3.1 Android**

Die native Programmiersprache für das Android Betriebssystem ist Java. Die Programmierung in Java bringt den Vorteil, dass auf die gesamte Application Programming Interface (API) von Android zugegriffen werden kann. Weiter sind die Geräte



Abbildung 2.1:  
Struktur der  
Applikation als  
Organigramm



genau dafür ausgelegt und die optimale Performance kann erreicht werden. Sämtliche Komponenten dieser Arbeit sind in Java geschrieben. Für die Persistierung der Daten auf dem Tablet wird eine SQLite Datenbank verwendet.

### 2.3.2 Externe Libraries

Android beinhaltet bereits ein umfangreiches Framework zur Entwicklung. Einzig beim Object-relational mapping (ORM), also beim Abbilden von Objektdaten in der Datenbank kommt eine externe Library zum Einsatz.

*ORMLite*<sup>1</sup> ist eine OpenSource Java Library, welche auch für Android eine optimale Lösung bietet. Die zu verwendenden Felder einer Klasse können mit Java Annotationen versehen werden, daraus versucht ORMLite dann die Datenbank zu beschreiben. In dieser Arbeit wurde die ORMLite Version 4.39 verwendet.

### 2.3.3 Entwicklungsumgebung

Die von Android empfohlene Entwicklungsumgebung ist Eclipse<sup>2</sup> mit einem Plugin zur Entwicklung von Android Applikationen. Auf der Entwicklerseite von Android steht dazu folgendes:

“Android Development Tools (ADT) is a plugin for the Eclipse IDE that is designed to give you a powerful, integrated environment in which to build Android applications.”<sup>3</sup>

Eclipse ist eine weit verbreitete IDE und wird aktiv weiter entwickelt. Mit dem Plugin zusammen bilden Sie eine solide Grundlage für dieses Projekt.

Damit die Android Applikation direkt auf dem Computer getestet werden kann, stellt Google ein Emulator zur Verfügung. Der Emulator ist allerdings auch als solcher zu betrachten da die Bedienung nicht vergleichbar ist mit einem richtigen Tablet.

### 2.3.4 Android Version

Eine Anwendung wird für eine spezifische Android Version entwickelt und getestet, somit kann garantiert werden, dass das Verhalten der Anwendung immer gleich ist.

1. ORMLite, <http://ormlite.com/>, Aufgerufen am 23.05.2012

2. Eclipse, <http://eclipse.org/>

3. Android Plugin für Eclipse, <http://developer.android.com/sdk/eclipse-adt.html>

In dieser Arbeit ist dies die Version 3.1 mit dem Versionsnamen *Honeycomb*.<sup>4</sup> Die Entwicklung auf einer Version schliesst jedoch nicht aus, dass die Anwendung in neueren Versionen nicht mehr lauffähig ist. Auch *RadioTour* kann für zukünftige Versionen weiterentwickelt und verwendet werden.

## 2.4 Mitbewerberanalyse

Die Art der Applikation ist sehr spezifisch und kann nicht direkt auf andere Sportereignisse angewendet werden. Deshalb beinhaltet die Analyse von Mitbewerbern nur die grossen europäischen Radrennen. Wie bei der Tour de Suisse ist auch in Frankreich an der *Tour de France*<sup>5</sup> ein RadioTour Speaker mit dabei. Darüber wie die Aufzeichnungen in Frankreich im genauen stattfinden kann aber nur spekuliert werden da die Informationen nicht öffentlich zugänglich sind.

In Italien findet zum Zeitpunkt dieser Arbeit der *Giro d'Italia*<sup>6</sup> statt. Bei diesem Radrennen ist es möglich aus den Informationen, welche auf der Webseite verfügbar sind, zu schliessen, dass ein ähnliches System verwendet wird. Während dem Rennen ist es möglich die aktuelle Rennsituation zu betrachten.

Abbildung 2.2:  
Rennsituation am  
Giro d'Italia



In der Abbildung 2.2 ist der Live Abschnitt der offiziellen Webseite zu sehen. Im oberen Teil wird der Standort in der aktuelle Etappe eingeblendet. Weiter unten ist die Situation an der Spitze abgebildet. Die Fahrer sind nach Rückstand gruppiert. Da jedoch nicht zu erkennen ist, wie die Informationen im Feld erfasst werden, muss die Mitbewerberanalyse an dieser Stelle abgeschlossen werden.

4. AndroidVersionHoneycomb, [http://de.wikipedia.org/wiki/Android\\_\(Betriebssystem\)#Versionsverlauf](http://de.wikipedia.org/wiki/Android_(Betriebssystem)#Versionsverlauf)

5. Tour de France, <http://www.letour.fr/>

6. Giro d'Italia, <http://www.gazzetta.it/Speciali/Giroditalia/2012/>

Im folgenden Abschnitt wird die Architektur der Applikation diskutiert. Die Architektur ist so gewählt, dass die einzelnen funktionalen Komponenten zueinander eine tiefe Abhängigkeit aufweisen und dadurch eine weitere Entwicklung möglichst einfach ist.

---

### 3.1 Klassendiagramm

Die Domainlogik beinhaltet die Kernelemente der Applikation. Einerseits sind dies die Rennfahrer, welche Informationen über sich festhalten andererseits die Etappe mit den Informationen zur Strecke. Während dem Rennen werden die Fahrer in Gruppen unterteilt. Auch diese Gruppen sind in der Domain abgebildet. Das Klassendiagramm des Domain Package zeigt die wesentlichen Elemente.

*Rider* speichert die Angaben zu einem Fahrer und beinhaltet keine eigene Logik. Ein Fahrer hat immer genau ein *RiderState*. Dies ist ein Java Enum und zeigt den Status des Fahrer an. Nach dem Import der Fahrerliste werden alle Fahrer auf *activ* gesetzt. In *Stage* ist die Etappe definiert. Jede Etappe hat eine Marschtabelle in Form von mehreren *PointOfRace* Objekten. Diese Objekte werden durch Import der Marschtabelle erstellt.

Da pro Etappe jeder *Rider* einen anderen *RiderState* haben kann, gibt es die Verbindungsklasse *RiderStageConnection*. In dieser Klasse ist jeweils die Etappe mit dem Fahrer verknüpft. Dies ermöglicht es den Rückstand eines Fahrers in mehreren Etappen zu verfolgen. Ein *Judgement*, also eine Wertung, gehört immer zu einer Etappe. Diese Wertungen sind definiert durch ein *SpecialRanking*, welche Punkte- und Zeitboni beinhalten können.

---

### 3.2 Sequenz Diagramm

Der häufigste UseCase besteht darin, die Rennsituation zu erfassen und an den Server zu übermitteln. Gleichzeitig werden im Hintergrund die Live Informationen aktualisiert. Diese beiden Hauptanwendungsfälle sind im folgenden System Sequenz Diagramm dargestellt. Der Aktor wird durch den *RadioTour Speaker* dargestellt und das System durch die *RadioTour* Applikation. Das externe System stellt die Serverseite dar.

Das Rennen wird durch das Starten der Rennzeit gestartet. Ab diesem Zeitpunkt beginnt die Aufzeichnung des Rennkilometers und die Berechnung der durchschnittlichen Geschwindigkeit.

Abbildung 3.1: Die  
Domainklassen in  
der Abhängigkeit

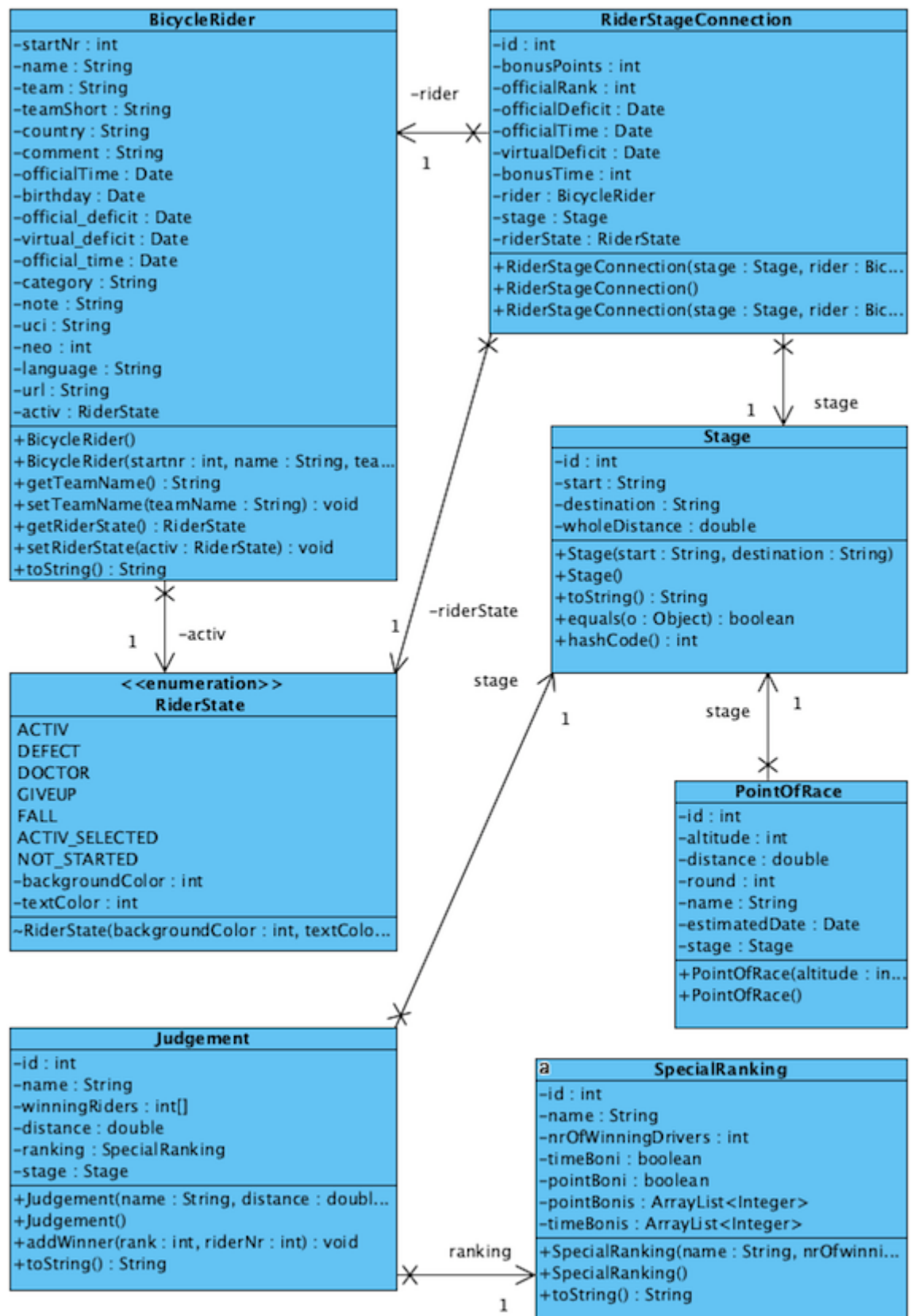
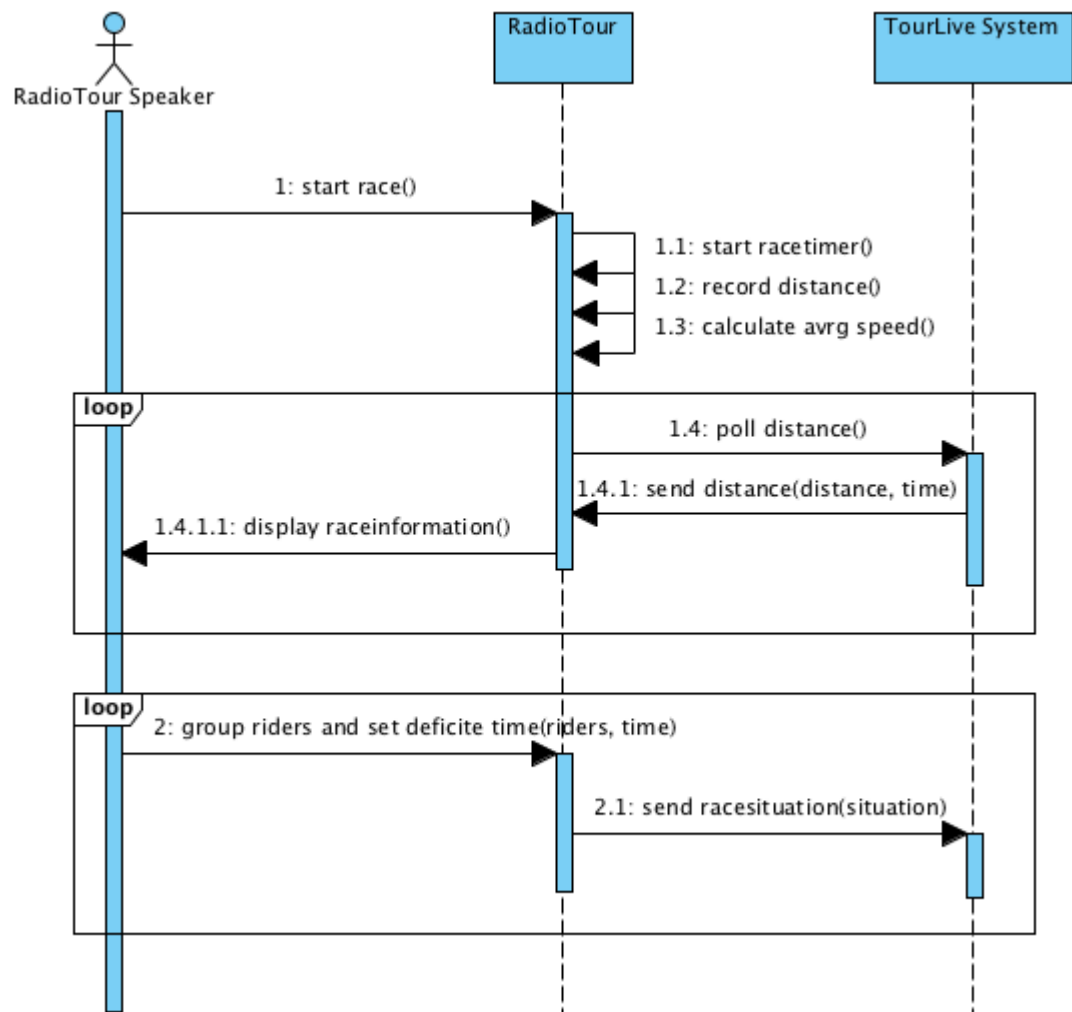


Abbildung 3.2: Das System Sequenz Diagramm





Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.





---

### 5.1 Java Unit Tests & Android Testing Framework

Da die Programmiersprache für Android Applikationen Java ist, sind Java Unit Test möglich. Android bietet weiter noch ein Testing Framework an, bei welchem gesamte UseCases getestet werden können. In dieser Arbeit gibt es keine vollständige Testabdeckung, da viel Arbeit im User Interface steckt und in der optimalen Anpassung an das Anwendungsumfeld. Dennoch gibt es folgende Testfälle:

- Testcase 1
- Testcase 2
- Testcase 3

---

### 5.2 Feldtest

Um die Benutzerfreundlichkeit und den Mehrwert der Applikation im Vergleich zur bisherigen Web Applikation zu ermitteln ist ein Feldtest unabdingbar. Es gab die Möglichkeit die *RadioTour* Applikation in einem Berner Fahrrad Rennen zu testen. Bei der Berner Rundfahrt<sup>1</sup> bot sich der *RadioTour Speaker* der Berner Rundfahrt, David Loosli<sup>2</sup> der Berner Rundfahrt an, die Applikation zu testen. In einem ersten Treffen wurden die grundlegenden Funktionen der Applikation und die Bedienung erklärt. Weiter ist ein Ausschnitt aus einem fiktiven Rennen durchgespielt worden, wobei eine Person den Chronofunk simulierte. Ein kurzer Auszug aus der vorbereiteten Situation ist unten aufgeführt.

- alle Fahrer sind importiert und das Rennen beginnt jetzt
- Rennzeit wird gestartet
- *Chronofunk*: Fahrer 4 & 17 von Beginn an, an der Spitze
- *Chronofunk*: Bereits 1:07 Vorsprung
- *Chronofunk*: 31 hat ein defektes Rad und muss raus
- *Chronofunk*: 8, 83 & 34 fallen hinter das Feld mit einem Rückstand von 4:31

---

1. Berner Rundfahrt <http://www.berner-rundfahrt.ch>

2. David Loosli, ehemaliger profi Radrennfahrer und *RadioTour Speaker*

Der vollständige Usability Test sowie der Testlauf an der *Berner Rundfahrt* mit den Rückmeldungen von Herrn Loosli sind im Anhang 8.8 zu finden.

Mit den Erfahrungen und den Rückmeldungen aus diesem Event konnte das Endprodukt massiv verbessert werden. Die wesentlichen Punkte die für die Weiterentwicklung verwendet wurden sind:

- TimePicker Nummern (Auswahl des Rückstandes einer Gruppe) sind zu klein (Korrektur siehe Abbildung 5.1)
- nur die Fahrer welche aufgegeben haben oder nicht erschienen sind sollen ausgegraut werden (Vorschlag in Abbildung 5.2)
- ein Fahrer kann folgende Status haben:
  - im Rennen / aktiv
  - nicht gestartet
  - ausgeschieden
- Rennzeit und Kilometer sind nach einem Absturz der Applikation noch verfügbar

Abbildung 5.1:  
TimePicker -  
Auswahl eines  
Zeitrückstandes

Wählen des Rückstandes der Gruppe relativ zur Spitze																			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19	10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29	20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39	30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49	40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59	50	51	52	53	54	55	56	57	58	59

Besonders zu hervorheben ist an dieser Stelle, dass Herr Loosli entgegen den Erwartungen die Fahrer, welche in einer Gruppe eingeteilt sind (z.B. Spitze) nicht ausgegraut haben möchte. So entstand der, in Abbildung 5.2 dargestellte Entwurf.

Die Abbildung 5.2 zeigt eine Situation, bei der die Fahrer 33 und 106 nicht gestartet sind und die Fahrer 152 und 154 aufgegeben haben. Diese Fahrer sind farblich markiert und durchgestrichen. Die weiteren Farben entsprechen den besonderen Ereignissen; Arzt, Sturz oder Defekt eines Fahrers bzw. eines Fahrrads.

Abbildung 5.2:  
RiderPicker -  
Auswahl von  
Fahrern

1	2	3	4	5	6	7	8	
11	12	13	14	15	16	17	18	
21	22	23	24	25	26	27	28	
31	32	33	34	35	36	37	38	
41	42	43	44	45	46	47	48	
51	52	53	54	55	56	57	58	
61	62	63	64	65	66	67	68	
71	72	73	74	75	76	77	78	
81	82	83	84	85	86	87	88	
91	92	93	94	95	96	97	98	
101	102	103	104	105	106	107	108	
111	112	113	114	115	116	117	118	
121	122	123	124	125	126	127	128	
131	132	133	134	135	136	137	138	
141	142	143	144	145	146	147	148	
151	152	153	154	155	156	157	158	
161	162	163	164	165	166	167	168	
171	172	173	174	175	176	177	178	
Arzt		Sturz		Defekt		Aufgabe		DNC
Auswahl aufheben								



---

# 6

## Ergebnisse und Schlussfolgerungen

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.



---

# 7

## Verzeichnisse und Referenzen

---

### 7.1 Literaturverzeichnis

---

### 7.2 Tabellenverzeichnis

---

### 7.3 Abbildungsverzeichnis





---

# Abbildungsverzeichnis

1.1	Das Aufgabenumfeld in einem BigPicture zusammengefasst . . . . .	11
2.1	Struktur der Applikation als Organigramm . . . . .	17
2.2	Rennsituation am Giro d'Italia . . . . .	18
3.1	Die Domainklassen in der Abhängigkeit . . . . .	20
3.2	Das System Sequenz Diagramm . . . . .	21
5.1	TimePicker - Auswahl eines Zeitrückstandes . . . . .	26
5.2	RiderPicker - Auswahl von Fahrern . . . . .	27
8.1	Das BigPicture in voller Grösse . . . . .	40
8.2	UseCase Diagramm . . . . .	43
8.3	Die Fahrerauswahlliste zur Gruppierung in der bisherigen Web Applikation . . . . .	43
8.4	Gruppierung in der bisherigen Web Applikation . . . . .	43
8.5	Die Spezialklasselemente und Wertungen in der bisherigen Web Applikation . . . . .	44
8.6	Virtuelles Klassement in der bisherigen Applikation . . . . .	45
8.7	Die Fahrerliste mit den Informationen zum Status der Fahrer . . . . .	45



**API**

Application Programming Interface. 16

**Chronofunk**

Die Motorradfahrer, welche im Rennfeld verteilt mitfahren und die Positionen der Ausreissergruppen per Funk an den RadioTour Speaker übermitteln.. 25

**Enum**

Ein Enum ist ein Datentyp mit fest bestimmten Konstanten. Es kann immer nur ein Wert ausgewählt sein. 19

**IDE**

Integrated Development Environment. 12, 17

**Java Unit Test**

Java bietet die Möglichkeit integrierte Softwaretests automatisiert durchzuführen. Die erleichtert die Arbeit enorm und unterstützt ein Entwicklungsteam, eine möglichst hohe Testabdeckung zu erarbeiten. first. 25

**ORM**

Object-relational mapping. 17

**RUP**

Rational Unified Prozess. 7, 8

**Splashscreen**

Eine Anzeige, die oftmals beim Start einer Applikation die Wartezeit bis zur vollständigen Initialisierung überbrückt.. 16

**SQLite**

Eine Datenbank in einer Datei first. 17



---

# Literaturverzeichnis

- [1] Robbie Matthews. *Beginning Android Tablet Programming (Beginning Apress)*. Apress, 1 edition, 2011.
- [2] Donn Felker. *Android Tablet Application Development For Dummies*. For Dummies, 1 edition, 2011.
- [3] Sven Riedel. *Git- kurz & gut*. O'Reilly, 1 edition, 2009.
- [4] Thomas Künne. *Android 3: Apps entwickeln mit dem Android SDK (Galileo Computing)*. Galileo Computing, 1 edition, 2011.
- [5] Wikipedia. Optimistic concurrency — Wikipedia, the free encyclopedia. [http://de.wikipedia.org/wiki/Optimistic\\_Concurrency](http://de.wikipedia.org/wiki/Optimistic_Concurrency), 2012. [Online; accessed 10-May-2012].
- [6] cnlab AG, Dr. Prof. Peter Heinzmann. Administratives zu studien- und bachelorarbeiten. <http://www.cnlab.ch/kurse/SABA/>, 2012. letzter Zugriff am 30.05.2012.
- [7] Andrew S. Tanenbaum and Maarten Van Steen. *Distributed systems: principles and paradigms*. Pearson Prentice Hall, 2007.
- [8] Todd Greanier. Discover the secrets of the java serialization api, 2000. [Online; accessed 17-May-2012].



## 8.1 BigPicture

Im folgenden ist das Aufgabenumfeld mit den äusseren Aktoren und Einflussbereiche in einem BigPicture zusammengefasst.

## 8.2 Projektmanagement

Für die Projektorganisation kommen verschiedene Anwendungen und Hilfsmittel zur Anwendung. Im folgenden werde die einzelnen Bereiche behandelt.

### 8.2.1 Zeitplanung

Die Zeiterfassung ist in einem dafür vorbereiteten Excel File abgelegt. Folgende Meilensteine sind im Zeitplan festgelegt:

Projektstart		21.02.2012
Milestone 1	RadioTour Bestandesaufnahme Webanwendung	05.03.2012
Milestone 2	Plattform Entscheid gefällt	12.03.2012
Milestone 3	Features und Abgrenzung definiert und mit Heinzmann besprochen	19.03.2012
Milestone 4	Prototyp "Rennsituation,, bei Heinzmann zum Testen übergeben	09.04.2012
Milestone 5	Dokumentation Teil Einleitung zur Review übergeben	23.04.2012
Milestone 5.1	Prototyp zum Testen an Heinzmann übergeben	01.05.2012
Milestone 6	Feature Freeze	07.05.2012
Rennen	Berner Rundfahrt, Lyss	12.05.2012
Milestone 7	Code Freeze	14.05.2012
Milestone 8	Dokumentation Freeze	21.05.2012
Milestone 9	Abgabe komplett	28.05.2012
Projektende		01.06.2012
Rennen	Gippingen (nach Abgabe)	07.06.2012
Rennen	Tour de Suisse (nach Abgabe)	09.-16.06.2012

Die Auswertung der Zeiterfassung zeigt deutlichen mehr (Ist-) Arbeitsstunden an, als die in der Planung eingetragenen (Soll-) Stunden. Besonders bei der Implementierung ist oftmals mehr Zeit benötigt worden als ursprünglich geplant war. Weiter hat das Erstellen der Dokumentation weitaus mehr Zeit in Anspruch genommen als erwartet. Der Projektzeitplan mit den vollständigen eingetragenen Arbeitsstunden ist

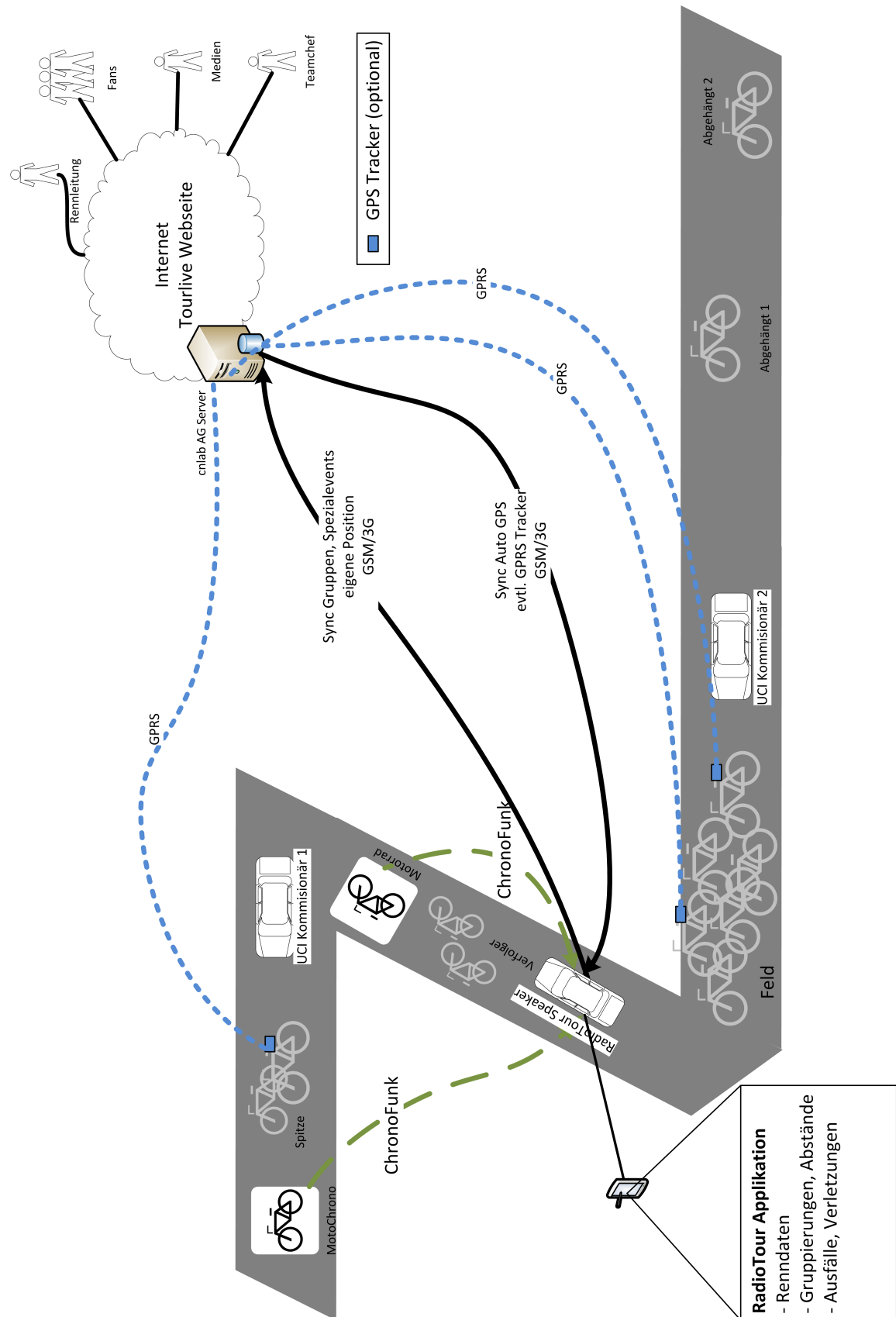


Abbildung 8.1: Das BigPicture in voller Grösse



in digitaler Form auf der beigelegten CD.

### 8.2.2 Code Base und Issue Tracking

Das Code Repository wurde auf Github <sup>1</sup> erstellt und verwaltet. Dieses Repository ist öffentlich lesbar jedoch kann nicht öffentlich darauf geschrieben werden.

Nach der ersten Implementierungsphase sind die Probleme, Fehler und weiteren Features der Applikation im integrierten Issuetracker von Github erfasst. Die Issues sind einer der folgenden Kategorie zugeordnet:

- must
- can
- nice

Sämtliche “must,, und “can,, Issues wurden erfolgreich abgearbeitet. Offen geblieben ist ein einziges “nice,, Issue, dabei geht es um die Auswahl von Fahrern für die Zuweisung von Wertungen in einem Spezialklassement.

---

## 8.3 Persönliche Berichte

### 8.3.1 Daniel Stucki

### 8.3.2 Florian Bentele

Zu Beginn des Projekts war es schwierig für mich den Arbeitsaufwand und die Erwartungen an mich abzuschätzen. Die Aufgabenstellung ist mehr als einen Anstoss zu verstehen und wird ständig weiter entwickelt und konkretisiert. Da wir aber schnell in die Thematik gefunden haben und unsere Ideen bei jeder Besprechung ein Stück weiter zu einem einsetzbaren Produkt führten, schwanden sämtliche Unsicherheiten. Besonders der Reiz, eine Applikation zu entwickeln, die den Anforderungen eines Live Sport Events entsprechen muss und in der Realität eingesetzt werden soll führten zu einer hohen Einsatzbereitschaft und grosser Motivation. Ich hatte eine gewisse Verantwortung übernommen, die Applikation am Ende des Semesters funktionstüchtig abzuliefern. Weiter war es mir ein Anliegen eine Arbeit im Bereich Mobile oder Web Applikationen zu erstellen. Mit diesem Projekt gelang es mir einen tiefen Einblick in die Entwicklung von Android Applikationen zu erarbeiten.

Die ersten Wochen verliefen dennoch harzig, da die Analyse der bestehenden Applikation sehr aufwändig und zeitintensiv war. Sämtliche Tools und das Projektmanagement wurden eingerichtet, die Aufgaben aufgeteilt. Bei der Implementierungsphase ging es dann sehr gut voran. Wir haben das Aufgabenumfeld in Teilprobleme aufgeteilt und diese dann Abgearbeitet. In weiteren Projekten würde ich mehr Wert auf diese Aufteilung legen. Nach der fundamentalen Architektur, die in meinen Augen ein Prozess ist, bei dem das ganze Team dabei sein sollte, können die Features gut aufgeteilt werden. Nach dem Hauptteil der Implementierung hatten wir die einmalige Gelegenheit, unsere Applikation in einem Radrennen zu testen. Dies ermöglichte uns ein echtes, unverfälschtes Feedback zu erhalten. Aus den Rückmeldungen konnten wir die Applikation weiter verbessern und für die Tour de Suisse bereit machen. Im weiteren Verlauf versuchten wir den bestehenden Code zu optimieren und verschönern (Refactoring). Da aber bis in der zweitletzten Woche des Projekts noch Features

---

1. Github, <https://github.com/dstucki/RadioTour>

hinzukamen, war dies sehr schwierig. Gegen Ende des Projekts kamen immer mehr Aufgaben im Bereich Dokumentation dazu. Aber auch diesen Teil konnten wir gut aufteilen.

Die Zusammenarbeit mit Daniel Stucki hat optimal funktioniert. Ich konnte viel von seiner Erfahrung als Java Entwickler profitieren und wir haben die Herausforderung gemeinsam gemeistert. Insbesondere in der Entwicklung ist es für mich wichtig, Teilprobleme und deren Lösungsansätze mit einer involvierten Person diskutieren zu können. Ich freue mich auch in der Bachelorarbeit mit Daniel zusammen zu arbeiten.

Aus diesem Projekt nehme ich diverse Erfahrungen mit, einerseits das Erlernte im Bereich der Android Programmierung andererseits aus dem Projektmanagement und der Wichtigkeit einer möglichst genauen Zeitplanung. Auch die Meilensteine und der Feldtest sind gute Methoden um den Projektverlauf möglichst Zeitgerecht einzuhalten.

---

## 8.4 Software Dokumente

---

## 8.5 Kriterienkatalog

Hier sind die Kriterien

---

## 8.6 Kaufempfehlung

---

## 8.7 UseCases der bisherigen Applikation

Im unten stehenden UseCase Diagramm (Abbildung 8.2) sind die primären UseCases aufgeführt. Nur der RadioTour Speaker erfasst Daten in dieser Applikation und ist daher der einzige Akteur. Das System wird durch die RadioTour Applikation abgebildet. Zur besseren Darstellung wurden einzelne UseCases vereinfacht oder zusammen gefasst.

- **Fahrer auswählen**

Dem RadioTour Speaker muss es möglich sein, einen oder mehrere Fahrer schnell auszuwählen. Die Fahrer werden im Auswahldialog bevorzugt durch ihre Startnummern dargestellt. An der Tour de Suisse besteht ein Team – nach Aussage von P. Heinzmann – aus 8 Fahrern. Um eine möglichst gute Übersicht zu gewährleisten werden die Fahrer jeweils Zeilenweise in deren Teams gruppiert. Ausgewählte Fahrer werden farblich hervorgehoben. Die Nummern der Fahrer, welche bereits Gruppen zugewiesen wurden, werden in Klammern dargestellt. Die Nummern ausgeschiedener Fahrer werden gestrichen dargestellt.

- **Fahrer gruppieren**

Dem RadioTour Speaker muss es möglich sein, die ausgewählten Fahrer in Gruppen zu organisieren. So kann er die ihm gemeldeten Rennsituationen mit Ausreissern, Verfolgern, Feld und abgehangen darstellen.

Abbildung 8.2:  
UseCase Diagramm

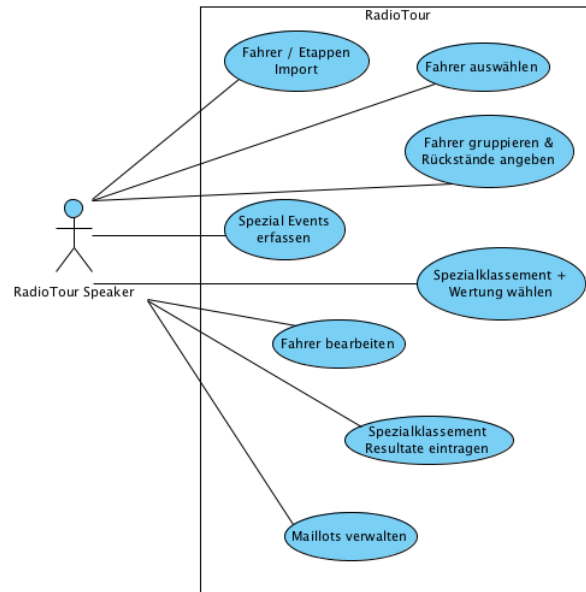


Abbildung 8.3: Die  
Fahrerauswahlliste  
zur Gruppierung in  
der bisherigen Web  
Applikation

set 3 Fahrer gewählt

1	2	3	4	5	6	7	8	
11	12	13	14	15	16	17	18	
21	22	23	24	25	26	27	28	
31	32	33	34	35	36	37	38	
41	42	43	44	45	46	47	48	
51	52	53	54	55	56	57	58	
61	62	63	64	65	66	67	68	
71	72	73	74	75	76	77	78	
81	82	83	84	85	86	87	88	
91	92	93	94	(95)	96	(97)	98	
101	102	103	104	(105)	106	107	108	
111	112	113	114	115	116	117	118	
121	122	123	124	125	126	127	128	
131	132	133	134	135	136	137	138	
141	142	(143)	144	145	146	147	148	

Abbildung 8.4:  
Gruppierung in der  
bisherigen Web  
Applikation

15:27:52, -1km Spitze->RadioTour: 0 (0km) Spitze->Feld: 0 (0km) Status: **RadioTour**

Ren Spe Vir Fah Adm Log Etappe 3 Distanz: 0 km

set

Spitze: 2 95 Alberto LOSADA ALG.. KAT [32]  
97 Luca PAOLINI KAT [83]

set

Rückstand: 2 105 Mathew HAYMAN SKY [136]  
143 Jorge AZANTA SOTO EUS [23]

set

Rückstand: set  
Feld Fahrer-Pickliste

Spe Vir Fah Pickliste-Fahrer

Arzt Aufgabe  
Defekt Start

- **Rückstände angeben**

Dem RadioTour Speaker muss es möglich sein, für die Gruppen (siehe oben) ihre jeweiligen Zeitabstände relativ zur Spitze einzugeben. Falls vorhanden,

sollen auch die mit dem TourLive GPS-System erfassten Zeitabstände Spitzefeld eingeblendet werden.

- **Spezial Events erfassen**

Dem RadioTour Speaker muss es möglich sein, für ausgewählte Fahrer Spezialereignisse festzulegen. Dies sind beispielsweise Arztbesuch, Aufgabe, Defekt oder einen Sturz.

Abbildung 8.5: Die Spezialklassemente und Wertungen in der bisherigen Web Applikation

The screenshot shows a web application interface for a cycling race. At the top, it says 'Etappe 3' and 'Distanz: 0 km'. Below this is a dropdown menu for 'Vaudoise Assurances GDM' and a link for 'Pickliste Gesamtklassement'. The main table is titled 'Grimselpass, 1. Kat. (50.1 km)'. It has columns for '#', 'Startnr', 'Bonus Punkte', and 'Fahrer'. The first five rows show the following data:

#	Startnr	Bonus Punkte	Fahrer
1	178	12	Wouter POELS (VCD)
2	76	8	Laurens TEN DAM (RAB)
3	7	6	Andy SCHLECK (LEO)
4	3	4	Jakob FUGLSANG (LEO)
5	121	2	Peter SAGAN (LIQ)
6			
7			
8			
9			
10			
11			

Two arrows point from text labels to the table: 'Spezialklassement' points to the 'Grimselpass, 1. Kat. (50.1 km)' title, and 'Wertung' points to the 'Bonus Punkte' column.

- **Spezialklassement und Wertung wählen**

Bei Mehretappenrennen werden typisch neben dem Gesamtklassement (schnellster Fahrer) mehrere Spezialklassemente (z.B. Bergpreis-, Sprintwertung, Punkteklasssement) gewertet. Innerhalb der Etappen gibt es jeweils mehrere Stellen (Wertungen), an denen für die Spezialklassemente Punkte vergeben werden. Dem RadioTour Speaker muss es möglich sein, die gewünschte Wertung zu einem der vorher erfassten Spezialklassemente auszuwählen.

- **Spezialklassement Resultate eintragen**

Dem RadioTour Speaker muss es möglich sein, für eine Wertung welche er ausgewählt hat (siehe UC oben), die Ränge zur Wertung mit Fahrernummern zu verbinden, wodurch das Klassement generiert wird.

- **Klassement anzeigen**

Das durch die eingetragene Wertung erstellte Klassement muss vom RadioTour Speaker abgerufen werden können. Dort sollen alle Fahrer angezeigt werden, welche einen Punkterang in diesem Spezialklassement erreichten.

- **Virtuelles Klassement**

Dem RadioTour Speaker muss es möglich sein, ein aktuelles Klassement der Tour abzurufen und dieses nach bestimmten Kriterien zu sortieren. Die zurzeit möglichen Sortierkriterien sind:

- Gruppen (zur Zeit des Aufrufs, nicht offiziell)
- Virtueller Rückstand (zur Zeit des Aufrufs, nicht offiziell)
- Zeitboni (zur Zeit des Aufrufs, nicht offiziell)
- Offizielle Zeit (zum Etappenende des Vortages, offiziell)
- Offizieller Rückstand (zum Etappenende des Vortages, offiziell)

Abbildung 8.6:  
Virtuelles  
Klassement in der  
bisherigen  
Applikation

Ren Spe Vir Fah Adm Log      Etappe 3      Distanz: 0 km

Rang	Startnr	Fahrername	Team	Land	Gruppen	virtuell Rückst.	Zeit- boni.	offiziell		
								Zeit		Rückst.
1	51	Damiano CUNEGO	Lampre - ISD	ITA		31:01:49 [3]		31:01:49	[1]	31:01:49
2	73	Steven KRUIJSWIJK	Rabobank Cycling Team	NED		31:03:25 [4]		31:03:25	[2]	31:03:25
3	1	Frank SCHLECK	Team Leopard-Trek	LUX		31:03:30 [5]		31:03:30	[3]	31:03:30
4	31	Levi LEIPHEIMER	Team RadioShack	USA		31:03:48 [6]		31:03:48	[4]	31:03:48

- **Fahrerliste anschauen**

Dem RadioTour Speaker muss es möglich sein, die aktuelle Fahrerliste anzuschauen. Die Fahrerliste ist nach Startnummer aufsteigend sortiert. (Die Startnummern werden in Mehretappenrennen so vergeben, dass die Fahrer eines Teams aufeinanderfolgende Startnummern erhalten.)

Abbildung 8.7: Die  
Fahrerliste mit den  
Informationen zum  
Status der Fahrer

Ren Spe Vir Fah Adm Log      Etappe 3      Distanz: 0 km

Startnummer	Fahrername	Team	Land
1	Frank SCHLECK	Team Leopard-Trek (LEO)	LUX
2	Fabian CANCELLARA	Team Leopard-Trek (LEO)	SUI
3	Jakob FUGLSANG	Team Leopard-Trek (LEO)	DEN

- **Fahrer de- bzw. aktivieren**

Dem RadioTour Speaker muss es möglich sein, einzelne Fahrer zu deaktivieren bzw. wieder zu aktivieren. Der Grund der Deaktivierung soll auch später noch nachvollziehbar sein. Es soll möglich sein, den Grund für die Deaktivierung anzugeben (z.B. ausgeschieden, nicht gestartet, andere). Eine so vorgenommene Deaktivierung eines Fahrers muss durch den RadioTour Speaker rückgängig gemacht werden können.

- **Fahrerdetails bearbeiten**

Dem RadioTour Speaker muss es möglich sein, einen Fahrer aus der Fahrerliste auszuwählen um seine Details anzuschauen und auch zu bearbeiten.

- **Statistik**

Dem RadioTour Speaker muss es möglich sein, in seinem Admin-Bereich eine kurze und prägnante textbasierte Statistik zu erhalten, bei welcher er auf einen Blick sieht wie viele Fahrer in der Datenbank sind und welche davon aktiv sind. Darüber hinaus die Anzahl Gruppen, Spezialklassemente, Wertungen und Vergebene Punkte.

- **Import Fahrerliste**

Nach jedem Renntag (= Etappe), wird in die RadioTour Applikation eine neue Fahrerliste mit den aktuellen offiziellen Zeiten importiert. Die Herausforderung besteht darin, dass das Format dieser Fahrerlisten im vornherein nicht bekannt ist. Deshalb muss es dem RadioTour Speaker möglich sein, den Importmodus noch dynamisch anzupassen. Derzeit sind für den Import der Daten verschiedene Importverfahren implementiert wie im Screenshot ersichtlich ist.

- **Import Spezialklassemente**

Dem RadioTour Speaker muss es möglich sein, Spezialklassemente zu importieren. Diese Importe werden vor der Tour de Suisse getätigt weshalb ein dynamischer Import hier nicht zwingend notwendig ist.

- **Speicherung Maillots**

Dem RadioTour Speaker muss es möglich sein, die Belegung der 4 verschiedenen Maillots anzugeben und zu sichern. Folgende Maillots gibt es:

Name	Farbe
Bergpreis	Rot-Weiss
Gesamtklassement	Gelb
Neo-Profi	Weiss
Punkte	Grün

- **Daten exportieren**

Dem RadioTour Speaker muss es möglich sein, die Spezialklassemente, Wertungen, Fahrer nicht nur zu erfassen, importieren und bearbeiten, sondern auch als \*.csv zu exportieren. Dabei werden einfach alle mit dem gewünschten Export assoziierten Infos in das exportierte \*.csv geschrieben.

---

## 8.8 Usability Test

Mit David Loosli, dem RadioTour Speaker an der Berner Rundfahrt 2012, wurde am 11.05.2012 einen Testlauf mit dem Tablet durchgeführt. Dabei wurde an einem vorgängigen Treffen folgende fiktive Situation durchgespielt, um den Umgang mit der Applikation zu erklären.

- alle Fahrer sind importiert und das Rennen beginnt jetzt
- Rennzeit wird gestartet
- Fahrer 4 & 17 von Beginn an, an der Spitze
- Bereits 1:07 Vorsprung
- 31 hat ein defektes Rad und muss raus
- der Vorsprung erhöht sich auf 2:19
- 8, 83 & 34 fallen hinter das Feld mit einem Rückstand von 4:31
- 8 kann wieder etwas aufholen und löst sich vom Feld, Rückstand nur noch 3:45
- 7 & 11 sind der spitze an den Versen mit einem Rückstand von 0:42
- 13 rückt zu 7&11 auf
- 41 ist verletzt und muss zum Arzt
- Chrono1 ist beim Ortseingang Lyss, ... Jetzt. Stoppuhr
- 37' 49' 45 rücken in die Verfolgergruppe vor
- wir sind beim Ortseingang Lyss, Stoppuhr

- 19 & 71 fallen eine Gruppe zurück
- 11 kommt an die Spitze
- Rennzeit wird korrigiert auf 05:11
- 71 kann nicht mehr und hat aufgegeben
- Rückstand vom Feld erhöht sich um 20 s
- Info kommt rein, dass Nr. 3 (Marcel Aregger) eigentlich Michael Aregger heisst
- Aregger (#3) kann auf die Spitze aufschliessen
- Hinter dem Feld bildet sich eine neue Gruppe mit den Fahrern 19, 24,
- 51 & 36. Rückstand 3:11
- 19 Stürzt
- 24 Rückt wieder ins Feld auf
- 51 und 36 fallen weiter in die nächste Gruppe zurück
- 52 rückt zur Spitze vor
- Rückstand Feld neu 1:30

#### **8.8.1 Auswertung und Feedback**

Alle Änderungen konnten durch David Loosli erfasst werden. Der erste Eindruck ist sehr gut, folgende Vorschläge wurden aufgenommen und in der Weiterentwicklung umgesetzt:

- Beim TimePicker sind die Nummern zu klein dargestellt, es benötigt keine Stunden
- Fahrer werden nur bei „Aufgabe“ und „nicht erschienen“ ausgegraut und durchgestrichen, bei „Arzt“, „Sturz“ und „Defekt“ werden die Fahrer Nummern eingefärbt
- ein Fahrer kann die folgenden Status haben
  - im Rennen / aktiv
  - nicht gestartet
  - ausgeschieden
- Tablet darf nicht verdunkeln falls es einen Moment nicht verwendet wird (Hardwareeinstellung)
- Rennkilometer muss editierbar sein, weiter sind die noch zu fahrenden Kilometer anzuzeigen
- Rennzeit und Kilometer sind nach einem Absturz der Applikation noch verfügbar