



HSR

HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz



Android Applikation RadioTour

Studienarbeit

Abteilung Informatik
Hochschule für Technik Rapperswil

Frühjahrssemester 2012
23. 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 anderen Motorfahrzeugen.

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 oder Stift ermöglicht es auf dem Tablet Radfahrer zu gruppieren und deren zeitliche Abstände zu erfassen. Aus diesen Angaben kann ein virtuelles Klassement erstellt werden. Diese live aufgezeichneten Informationen werden über das Mobilfunknetz auf die Server der TourLive Webseite übertragen und für die Medien und die Zuschauer aufbereitet.

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.

Inhaltsverzeichnis

0.1	Ziel	7
0.2	Abgrenzung	7
0.3	Erklärung	7
0.4	Ausgangslage	9
0.4.1	Live Informationen	9
0.4.2	Äussere Bedingungen	9
0.5	Vorgehensweise	9
0.6	Ergebnisse	10
0.7	Ausblick	10
1	Einleitung	11
1.1	BigPicture	11
1.2	Evaluation und Kaufempfehlung	12
2	Analyse	15
2.1	Struktur der Applikation	15
2.2	Technologien	15
2.2.1	Android	15
2.2.2	Externe Libraries	16
2.2.3	Entwicklungsumgebung	16
2.2.4	Android Version	16
2.3	Mitbewerberanalyse	16
3	Architektur	19
3.1	Domainmodel	19
4	Realisierung	21
5	Testing	23
6	Ergebnisse und Schlussfolgerungen	25

7	Verzeichnisse und Referenzen	27
7.1	Literaturverzeichnis	27
7.2	Tabellenverzeichnis	27
7.3	Abbildungsverzeichnis	27
	Glossar	31
I	Anhang	35
7.4	Projektmanagement	37
7.5	Software Dokumente	37
7.6	Kriterienkatalog	37
7.7	Kaufempfehlung	37

Aufgabenstellung

0.1 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.

0.2 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 ¹. Eine Übersetzung ist jedoch nicht Teil der Arbeit.

0.3 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

0.4 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.

0.4.1 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.

0.4.2 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.

0.5 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.²

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.

0.6 Ergebnisse

Die RadioTour Android Applikation beinhaltet die festgelegten Anforderungen. Für den Einsatz an der Tour de Suisse ist das Gerät bereit. Zu Beginn einer Etappe wird die aktuelle Fahrerliste sowie die Marschzeitabelle importiert. Der RadioTour Speaker kann die Veränderungen im Feld direkt auf dem Tablet erfassen. Die Bedienung ist wesentlich flüssiger und einfacher im Vergleich zur bisherigen Web Applikation.

0.7 Ausblick

Nach dem Einsatz an der Tour de Suisse müssen die Eindrücke und das Feedback des RadioTour Speakers aufgenommen werden. Diese Informationen dienen zur Verbesserung und zur weiteren Entwicklung des Produkts.

TODO - Abschnitt unfertig

2. Frei nach http://de.wikipedia.org/wiki/Rational_Unified_Proces

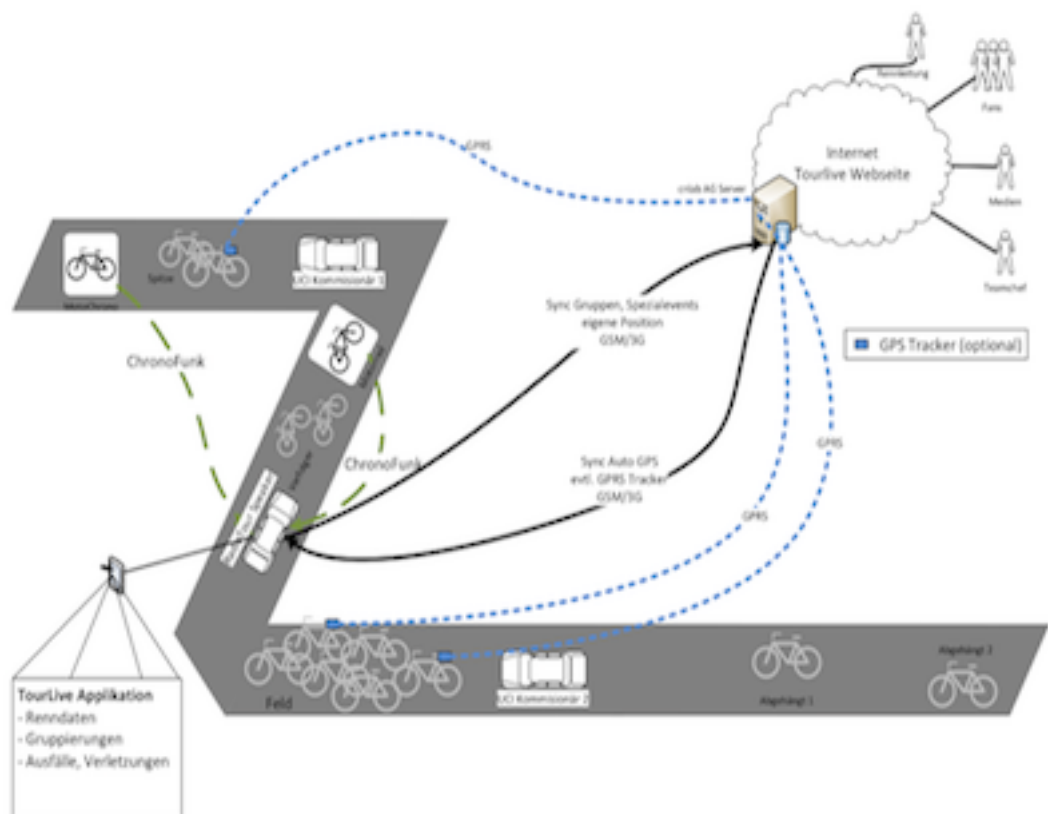
Im folgenden Abschnitt werden die aus technischer Sicht relevanten Aspekte genauer analysiert. Zu Beginn wird das Aufgabenumfeld in einem weiteren Sinne betrachtet, später gehen wir konkret auf die Analyse und das Software Engineering ein.

Der Hauptteil richtet sich vor allem an Personen, die bereits Hintergrundwissen zum Betriebssystem Android vorweisen sowie für Entwickler, die an der Weiterentwicklung des Produktes interessiert sind.

1.1 BigPicture

Zur Übersicht wird das Umfeld der Applikation in einem BigPicture zusammengefasst. Es ermöglicht die Darstellung der äusseren Einflussfaktoren sowie die Abgrenzung des Systems zu definieren.

Abbildung 1.1:
BigPicture



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 Tablett 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 markante 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.

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 und iOS. 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 7.6.

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 7.7 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 10.1¹

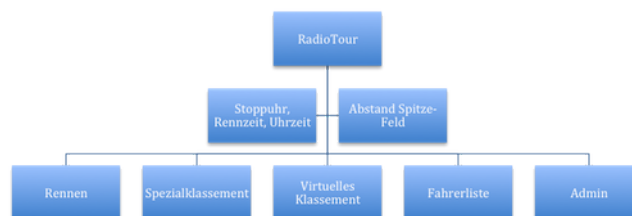
1. <http://www.samsung.com/ch/consumer/mobile-phone/tablets/tablets/GT-P7500UWDITV> aufgerufen am 23.05.2012

Als Grundlage für die RadioTour Applikation dient aus Sicht der Funktionalität die bestehende Web Applikation. Daraus werden die Requirements und UseCases erzeugt. Im folgenden werden die einzelnen Teile der Analyse vorgestellt.

2.1 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. So entsteht die Baumartige Struktur, wie sie in der Abbildung zu sehen ist.

Abbildung 2.1:
Struktur der
Applikation



2.2 Technologien

2.2.1 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 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.

2.2.2 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*¹ 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 der RadioTour Anwendung konnten alle Felder abgebildet werden.

2.2.3 Entwicklungsumgebung

Die von Android empfohlene Entwicklungsumgebung ist Eclipse² 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.“³

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.

2.2.4 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. In dieser Arbeit ist dies die Version 3.1 mit dem Versionsnamen *Honeycomb*.⁴

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.

2.3 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*⁵ 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*⁶ 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.

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>

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/>

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 aktuellen 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.

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

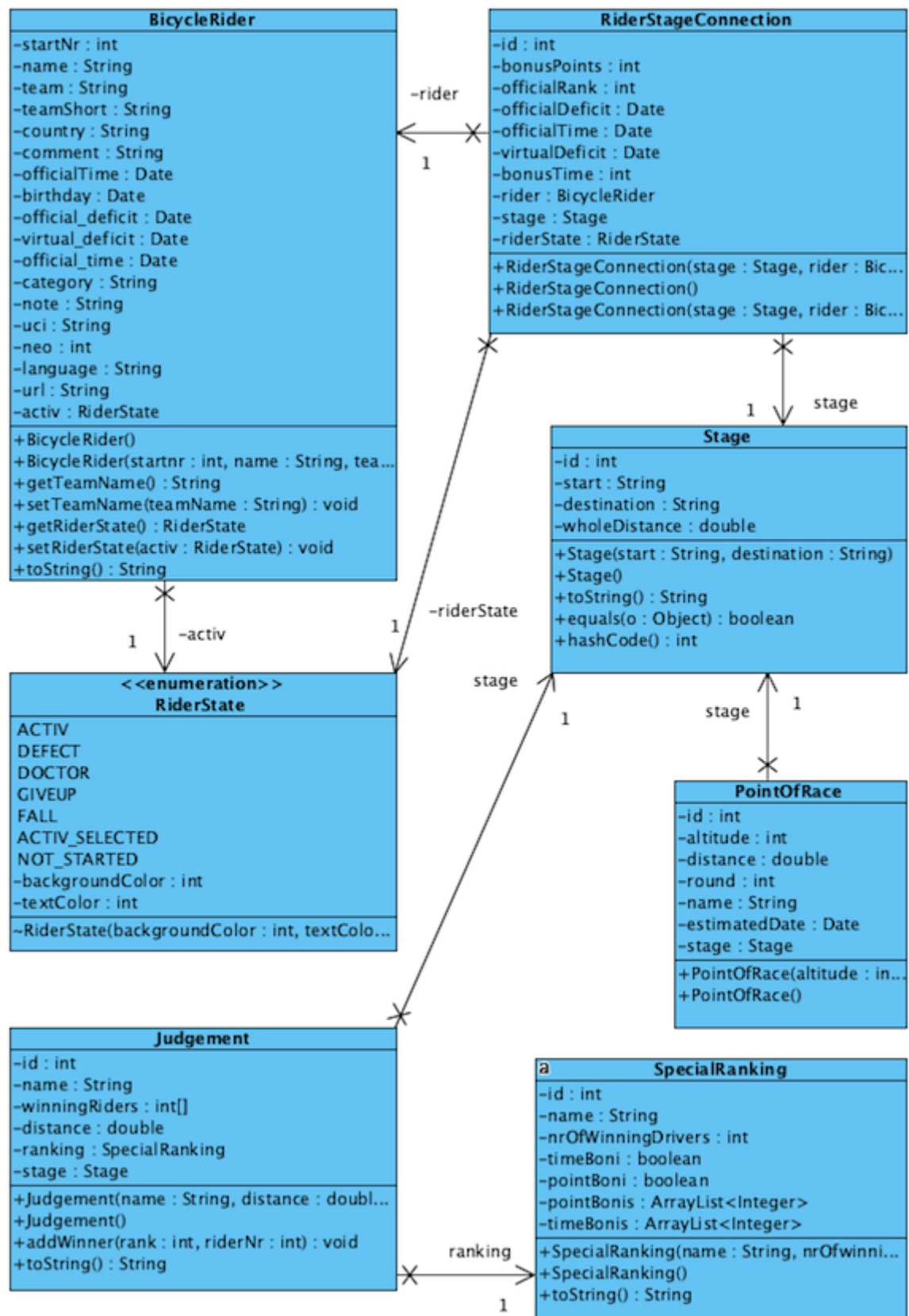
3.1 Domainmodel

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.

BicycleRider 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 *BicycleRider* 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.

Abbildung 3.1: Die
Domainklassen in
der Abhängigkeit



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.

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.

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	BigPicture	11
2.1	Struktur der Applikation	15
2.2	Rennsituation am Giro d'Italia	17
3.1	Die Domainklassen in der Abhängigkeit	20

API

Application Programming Interface. 15

Enum

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

IDE

Integrated Development Environment. 12, 16

ORM

Object-relational mapping. 16

RUP

Rational Unified Prozess. 9, 10

SQLite

Eine Datenbank in einer Datei. 15

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, 2012. [Online; accessed 10-May-2012].
- [6] Andrew S. Tanenbaum and Maarten Van Steen. *Distributed systems: principles and paradigms*. Pearson Prentice Hall, 2007.
- [7] Todd Greanier. Discover the secrets of the java serialization api, 2000. [Online; accessed 17-May-2012].

Teil I

Anhang

7.4 Projektmanagement

7.5 Software Dokumente

7.6 Kriterienkatalog

Hier sind die Kriterien

7.7 Kaufempfehlung