

# ROW@RE

Integriertes EDV-System zur Abwicklung von  
Ruderegatten

---

# Spezifikation

Version 0.11

14. April 2003

Oliver Urschel\*

Tammo van Lessen†

Joachim Praetorius‡

---

\*Aachen/Bad Kreuznach, <urschcom@users.berlios.de>

†Stuttgart/Mainz, <vanto@users.berlios.de>

‡Berlin, <jpraetorius@users.berlios.de>

© Copyright 2003 row@re team

Die Verteilung dieses Dokuments in elektronischer oder gedruckter Form ist gestattet, solange sein Inhalt einschließlich Autoren- und Copyright-Angabe unverändert bleibt und die Verteilung kostenlos erfolgt, abgesehen von einer Gebühr für den Datenträger, den Kopiervorgang usw.

Die in dieser Publikation erwähnten Software- und Hardware-Bezeichnungen sind in den meisten Fällen auch eingetragene Warenzeichen und unterliegen als solche den gesetzlichen Bestimmungen.

DRAFT

Dieses Dokument wurde mit  $\text{\LaTeX}$  gesetzt.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>5</b>
1.1	Zielsetzung . . . . .	5
1.2	Produktziele . . . . .	5
1.3	Produktumgebung . . . . .	5
1.4	Produktfunktionen . . . . .	5
<b>2</b>	<b>Grobentwurf</b>	<b>9</b>
2.1	Übersicht über das System . . . . .	9
2.1.1	Ausschreibung & Meldeschluß . . . . .	9
2.1.2	Startnummernvergabe - Regattaende . . . . .	10
<b>3</b>	<b>Anforderungen</b>	<b>11</b>
3.1	Aktoren . . . . .	11
3.1.1	Benutzer . . . . .	11
3.2	Use-Cases . . . . .	11
3.2.1	Am System anmelden . . . . .	12
3.2.2	Als Gast anmelden . . . . .	12
3.2.3	Ausschreibung erstellen . . . . .	13
3.2.4	Ausschreibung veröffentlichen . . . . .	14
3.2.5	Daten aus xregatta Format einlesen . . . . .	15
3.2.6	Daten eingeben . . . . .	16
3.2.7	Daten überprüfen und korrigieren . . . . .	16
3.2.8	Rennen verlosen . . . . .	17
3.2.9	Rennzeiten festlegen . . . . .	18
3.2.10	Meldeergebnis veröffentlichen . . . . .	19
3.2.11	Nachmeldung erfassen . . . . .	20
3.2.12	Abmeldung erfassen . . . . .	20
3.2.13	Ummeldung erfassen . . . . .	21
3.2.14	Jugendlizenzen erfassen . . . . .	21
3.2.15	Jugendlizenzen kontrollieren . . . . .	22
3.2.16	Schiedsrichtereinsatzplan erstellen . . . . .	22
3.2.17	Schiedsrichtereinsatzplan veröffentlichen . . . . .	23
3.2.18	Waage erfassen . . . . .	23
3.2.19	Kassenbuchungen erstellen . . . . .	24
3.2.20	Quittung erstellen . . . . .	24

3.2.21	Berichte erstellen . . . . .	25
3.2.22	Rennen verlegen . . . . .	25
3.2.23	Einsprüche erfassen . . . . .	25
3.2.24	Schiedsrichterbericht erstellen . . . . .	26
3.2.25	Urkunden drucken . . . . .	27
3.2.26	Ergebnisse veröffentlichen . . . . .	28
3.2.27	Startberechtigung kontrollieren . . . . .	28
3.2.28	Ausschluss von Mannschaften erfassen . . . . .	29
3.2.29	Mannschaften zu Rennen einteilen . . . . .	29
3.2.30	Rennen starten . . . . .	29
3.2.31	Zwischenzeiten nehmen . . . . .	30
3.2.32	Unfälle erfassen . . . . .	30
3.2.33	Einsprüche erfassen . . . . .	30
3.2.34	Zeit nehmen . . . . .	31
3.2.35	Zielfotos hochladen . . . . .	31
3.2.36	Rennzeiten einsehen . . . . .	31
3.2.37	Neue Nutzer anlegen . . . . .	32
3.2.38	Nutzer bearbeiten . . . . .	32
3.2.39	Nutzer löschen . . . . .	33
<b>4</b>	<b>Nichtfunktionale Anforderungen</b>	<b>35</b>
4.1	Leistungsanforderungen . . . . .	35
4.2	Entwurfseinschränkungen . . . . .	35
4.3	Verfügbarkeit . . . . .	35
4.4	Sicherheit . . . . .	35
4.5	Robustheit . . . . .	35
4.6	Wartbarkeit . . . . .	35
4.7	Portabilität . . . . .	35
<b>5</b>	<b>Graphische Benutzeroberfläche</b>	<b>37</b>
5.1	Anmeldefenster . . . . .	37
5.2	Hauptfenster . . . . .	37
<b>6</b>	<b>Daten</b>	<b>39</b>
<b>7</b>	<b>Begriffslexikon</b>	<b>41</b>
<b>8</b>	<b>Offene Fragen</b>	<b>43</b>

# 1 Einleitung

## 1.1 Zielsetzung

In diesem Dokument sind die Anforderungen an das Projekt **ROW@RE** detailliert spezifiziert. Das Produkt wird streng nach den hier aufgeführten Anforderungen entwickelt. Der Entwicklungsprozess gilt als erfolgreich abgeschlossen, wenn das Produkt allen hier aufgeführten Aspekten gerecht wird. Änderungen an diesem Dokument sind nach Abschluss der Spezifikationsphase nicht mehr ohne das Einverständnis des **ROW@RE** -Teams möglich.

## 1.2 Produktziele

Bei **ROW@RE** handelt es sich um ein Programmsystem, das alle Phasen einer Regatta unterstützt. Dazu gehört im Vorfeld der Regatta die Erstellung einer Ausschreibung und der Export dieser Informationen in verschiedene Medien. Des weiteren bietet **ROW@RE** Unterstützung für die Erfassung der eingegangenen Meldungen, die Durchführung der Startverlosung und das Erstellen des Meldeergebnisses. Während der Regatta unterstützt es das Regattabüro, Waage und Kasse bei Bezahlung, Um- und Abmeldungen, Einsprüchen und der Erfassung der Rennergebnisse. Nach der Regatta können sowohl automatisch korrigierte Meldeergebnisse, als auch Ergebnisse mit Zeiten und Plazierungen der Teilnehmer erstellt werden.

## 1.3 Produktumgebung

Das Programm soll auf beliebigen Plattformen lauffähig sein. Deshalb wird es als Webanwendung in PHP implementiert. Der Benutzer benötigt nur einen modernen Browser mit JavaScript-Unterstützung.

*Vielleicht sollten wir hier ein wenig konkreter sein. Javascript kann relativ unlustig sein. Ich denke da an tatsächliches ECMA-262 und W3C-DOM Unterstützung, was den Zugriff auf HTML-Elemente dann einheitlich über getElementByName() ermöglicht.*

## 1.4 Produktfunktionen

Das Produkt stellt dem Regattaveranstalter folgende Funktionen zur Verfügung

### Ausschreibung

- Ausschreibung erstellen

## 1 Einleitung

- Ausschreibung veröffentlichen (HTML, PDF, xregatta)

### Meldeschluss

- Daten einlesen aus xregatta-format
- Daten eingeben (Papiermeldung)
- Daten überprüfen und korrigieren
- Rennen verlosen
- Rennzeiten festlegen
- Meldeergebnis veröffentlichen (HTML, PDF, xregatta)

### Regattabüro

- Nachmeldung/Abmeldung/Ummeldung erfassen und damit verbundene Rennverschiebungen verarbeiten
- Jugendlizenzen kontrollieren/erfassen
- Lizenzliste einlesen / alternativ -> kontrollierte Lizenzen speichern (für weitere Veranstaltungen)
- Schiedsrichtereinsatzplan erstellen, veröffentlichen
- Waage erfassen
- Kasse Buchungen, Quittung erstellen, Berichte erstellen
- Rennen verlegen
- Einsprüche erfassen
- Schiedsrichterbericht erstellen
- Urkunden drucken
- Ergebnisse veröffentlichen (HTML, PDF, xregatta)

### Start/Ziel

- Startberechtigung kontrollieren (Übergewicht, Zusatzgewicht, unerlaubte Renngemeinschaften usw.)
- Ausschluss von Mannschaften erfassen (Geldstrafen???)
- Mannschaften zu Rennen einteilen

- Rennen starten
- Zwischenzeiten nehmen
- Unfälle erfassen (Kentern, Bootsschaden usw.)
- Einsprüche erfassen
- Zeit nehmen (Ziel)
- Zielfotos hochladen

### Informationen

- Besuchern Renninformationen zur Verfügung stellen.

DRAFT



## 2 Grobentwurf

Im Grobentwurf soll ein –absichtlicherweise– grobes Gerüst des Gesamtsystems entworfen werden. Ziel ist es den Rahmen, in dem die Use-Cases verwirklicht werden sollen abzustecken. Dies geschieht nicht um Planung und Implementation zu vermischen, was im Design nicht geschehen soll, sondern um einige Implikationen, die einige der Funktionalitäten voraussetzen durch die Struktur des Systems zu verwirklichen zu können. So darf es bestimmte Funktionen erst geben, wenn der Meldeschluß vorbei ist (z.B. Nachmeldung, Verteilung auf Läufe... ). Diesen Zeitpunkt kann man mit einer Strukturierung in Module erledigen, da der Gebrauch eines bestimmten Moduls eben einen bestimmten Zustand des Gesamtsystems annimmt.

### 2.1 Übersicht über das System

Das System soll in Modulen strukturiert sein. Dadurch sollen die wichtigen Zeitpunkte im “Lebenszyklus” einer Regatta voneinander abgegrenzt werden. Diese sind<sup>1</sup>:

- Ausschreibung der Regatta (Danach kann keine Rennen gelöscht werden)
- Meldeschluss (Meldungen danach sind Nachmeldungen)
- Startnummernvergabe (Danach kann keine Startnummern geändert werden)
- Meldeergebnis (Danach kann keine Mannschaft gelöscht werden)
- Regattabeginn (Aktivierung von regattaspezifischen Nutzerrechten)
- Rennbeginn (Danach kann nicht für das Rennen nach-,um- und abgemeldet werden)
- Rennabbruch (Führt evt. zum zeitversetzten Start des Rennens)
- Rennende (Danach kann die Teilnehmerliste nicht geändert werden)
- Regattaende
- TODO: Noch mehr wichtige Punkte ??

#### 2.1.1 Ausschreibung & Meldeschluß

Ausschreibung und Meldeschluß werden durch ein gemeinsames Modul 'Regattaverwaltung' behandelt. Hier können Regatten erzeugt und bearbeitet werden. Ebenso wird hier die Ausschreibung veröffentlicht. Ist dies geschehen, so kann die Regatta im nächsten Modul bearbeitet werden.

---

<sup>1</sup>Die Punkte stammen aus der Mail von Kenneth Drexel und sind nur leicht modifiziert.

### 2.1.2 Startnummernvergabe - Regattaende

Die übrige Funktionalität wird in einem zweiten Modul verwaltet. Hierbei wird durch die Nutzerverwaltung sichergestellt, daß jeder Nutzer nur die Aufgaben erfüllen kann, für die er vorgesehen ist<sup>2 3</sup>. Dadurch können bestimmte Abhängigkeiten untereinander konsistent im selben Modul behandelt werden, was eine Kommunikation der Module untereinander unnötig macht und somit Probleme von dieser Seite vermeidet. Die weiteren Konsistenzbedingungen (Start von Rennen etc.) können dann über die Datenbank abgehandelt werden.

DRAFT

---

<sup>2</sup>Alternativ zu diesem Zwei-Modul viele Nutzer Ansatz kann man natürlich auch in viele kleine Module untergliedern, die im Prinzip den Nutzerrollen entsprechen. Der Zugang zu diesen Modulen ist dann entsprechend per Passwort gesichert. Was davon vorteilhafter ist weiß ich nicht, vielleicht ist auch ein Implementierungsdetail, daß man die Nutzer in das entsprechende Modul weiterleitet, daß für sie vorgesehen ist. Hier ist eine Diskussion angebracht.

<sup>3</sup>Die Direktive 'jeder nur das, was er unbedingt muß' legt nahe auch noch feinere Nutzerrollen einzuführen wie 'Drucken', 'Meldungen' statt des allmächtigen Admin (der trotzdem existieren sollte). Auf eine Anpassung der Use-Cases habe ich allerdings verzichtet.

## 3 Anforderungen

Die Anforderungen werden in Form von Anwendungsfällen (Use-Cases) erklärt. Sie beschreiben die Funktionalität des zu erstellenden Systems.

### 3.1 Akteure

#### 3.1.1 Benutzer

Vor der Anmeldung an das System sind haben alle Benutzer die gleiche Funktion und die gleichen Möglichkeiten (**AKT Benutzer**). Nach der Anmeldung ergibt sich durch das Rechtemanagement eine neue Situation, die Benutzer werden in weitere Gruppen mit unterschiedlichen Rechten aufgeteilt.

- Systemverwalter (Einrichtung von **ROW@RE** , Verwalten der Nutzer) (**AKT Sysadmin**)
- Benutzer bei Regattaverwaltung (Ausschreibung, Erfassung Meldungen, Ergebnisse) (**AKT Regatta**)
- Benutzer im Regattabüro (**AKT Admin**)
- Benutzer an der Kasse (**AKT Kasse**)
- Benutzer an der Waage (**AKT Waage**)
- Benutzer am Start (**AKT Start**)
- Benutzer im Ziel (**AKT Ziel**)
- Benutzer an der Strecke (**AKT Zeit**)
- Gast (**AKT Gast**)
- TODO: weitere Benutzer?

### 3.2 Use-Cases

Die hier beschriebenen Geschäftsprozesse beschreiben ausschließlich die Interaktion zwischen Benutzer und Maschine. Implementations- und Gestaltungsdetails sind hier falsch.

#### 3.2.1 Am System anmelden

**Vorbedingung**

Browser ist gestartet und zeigt auf <http://ip-adresse/>

**Beschreibung**

Der Benutzer trägt seinen Namen und sein Passwort in Textfelder ein. Danach bestätigt er mit *Return* oder klickt auf *OK*. Darauf hin wird das Passwort überprüft.

**Nachbedingung Erfolg**

Der Benutzer ist nun angemeldet, es erscheint das Hauptfenster, die Auswahlen, die der Nutzer treffen kann sind seiner Nutzerrolle (siehe Aktoren) angepasst

**Nachbedingung Fehlschlag**

Es wird die Fehlermeldung "Falsches Kennwort" angezeigt, das Anmeldefenster bleibt angezeigt.

**Interaktionen**

AKT Benutzer

#### 3.2.2 Als Gast anmelden

**Vorbedingung**

Browser ist gestartet und zeigt auf <http://ip-adresse/>

**Beschreibung**

Gäste der Veranstaltung können Informationen über das Regattageschehen bekommen. Insbesondere können sie Zeiten zu Rennen einsehen. Hierfür existiert auf der Anmeldeseite ein Link "als Gast anmelden". Danach ist der als AKT Gast angemeldet. Diese Nutzerrolle hat keine Rechte.

**Nachbedingung Erfolg**

Der Nutzer ist als AKT Gast angemeldet. In dieser Rolle kann der Zeiten von beendeten Rennen einsehen.

**Nachbedingung Fehlschlag**

Das Anmeldefenster wird angezeigt.

**Interaktionen**

AKT Benutzer

### 3.2.3 Ausschreibung erstellen

#### **Vorbedingung**

Der Benutzer ist als AKT Regatta angemeldet

#### **Beschreibung**

Der Benutzer kann eine neue Regatta anlegen. Für diese kann er global Daten wie Termin, Austragungsort, Meldeschluß und Preise für die Rennen festlegen. Genauso sind die Kontaktdaten für die Meldung usw. anzugeben. Hierbei können vorhandene Personendaten aus einem Datenspeicher übernommen werden. Noch ungeklärte Daten müssen nicht sofort angegeben werden sondern können zu einem späteren Zeitpunkt ergänzt werden. Zur Identifizierung solcher Daten ist ein bestimmter Wert z.B. "-" vorzusehen. Danach kann der Nutzer beliebig viele Rennen anlegen, wobei zur Spezifizierung die folgenden Daten angegeben werden müssen: Bootsklasse, Leistungsklasse, Altersgruppe, Strecke. Ein mehrfaches Anlegen eines Rennens mit den *gleichen* Daten ist nicht möglich. Die Reihenfolge der Rennen ist zu jeder Zeit veränderbar. Die Rennen unter der Stelle, an der ein verschobenes Rennen eingefügt wird, werden nach unten verschoben. Ein nachträgliches bearbeiten der eingegebenen Daten ist zu jedem Zeitpunkt möglich. Die obigen Plausibilitätskontrollen werden jeweils neu getestet, so daß das System nur konsistente, sinnvolle Daten speichern kann.

#### **Nachbedingung Erfolg**

Die Veranstaltung wird mit den eingegebenen Daten gespeichert.

#### **Nachbedingung Fehlschlag**

Der Nutzer wird auf die Fehler in den eingegebenen Daten hingewiesen. Die Veranstaltung wird **nicht** gespeichert.

#### **Interaktionen**

AKT Regatta

#### 3.2.4 Ausschreibung veröffentlichen

##### **Vorbedingung**

Der Benutzer ist als AKT Regatta angemeldet. Es existiert eine Ausschreibung ohne inhaltliche Fehler. Alle Veranstalterdaten der Ausschreibung sind ausgefüllt und nicht mit "-" gefüllt.

##### **Beschreibung**

Eine abgeschlossene Ausschreibung kann durch den Nutzer veröffentlicht werden. Hierbei führt das System letzte Plausibilitätskontrollen an den eingegebenen Daten durch. Insbesondere müssen alle Daten die allgemein für die Regatta gelten ausgefüllt worden sein. Dann exportiert das System die relevanten Daten in verschiedene vom Nutzer wählbare Formate. Als Exportformate werden hierbei PDF, HTML und xregatta unterstützt. Diese Daten werden auch auf dem Server gehalten und können durch den Nutzer mittels des Browsers heruntergeladen werden. Eine veröffentlichte Ausschreibung kann nicht mehr verändert werden. Auf diesen Umstand soll der Nutzer vor der Veröffentlichung hingewiesen werden und sie explizit bestätigen müssen. Eine erneute Veröffentlichung in beliebigen der erwähnten Formate ist hingegen jederzeit möglich.

##### **Nachbedingung Erfolg**

Die Ausschreibung wird in den vom Nutzer gewählten Formaten veröffentlicht (Die Daten werden im passenden Format serialisiert). Eine Änderung der Ausschreibung ist nicht mehr möglich.

##### **Nachbedingung Fehlschlag**

Der Nutzer wird auf eventuelle Fehler in der Ausschreibung hingewiesen, die die Veröffentlichung verhindert haben (dazu gehört auch ein Abbruch des Nutzers bei dem Hinweis auf Unveränderbarkeit einer veröffentlichten Ausschreibung). Die Ausschreibung ist weiterhin änderbar, die Ausschreibungsdateien werden nicht erzeugt.

##### **Interaktionen**

AKT Regatta

### 3.2.5 Daten aus xregatta Format einlesen

#### **Vorbedingung**

Der Nutzer ist als AKT Admin angemeldet. Es existiert eine veröffentlichte Ausschreibung. Es liegt eine Meldedatei im xregatta Format vor.

#### **Beschreibung**

Der Nutzer kann aus einer Auswahl die Regatta auswählen, die er zu bearbeiten wünscht. Hierbei werden nur veröffentlichte Ausschreibungen berücksichtigt. Der Nutzer kann nun auswählen, daß er Meldedaten hinzufügen will. Hierzu existiert die Option "aus xregatta Format". Dazu kann der Nutzer über seinen Browser die entsprechende Datei auf den Server uploaden. Dort wird die Datei auf Validität gegenüber der DTD oder dem Schema getestet. Ist sie korrekt werden dem Nutzer die Daten aus der Datei angezeigt. Nun kann er diese per Klick für die Regatta übernehmen (*Vielleicht sollte man auch die Möglichkeit zulassen nur Teile zu übernehmen, also eine Tabellenansicht, in der man einzelne zeilen per Checkbox auswählen kann*). Für Aktive, deren Lizenz bei den Daten der Jugendlizenzen vorhanden sind wird ein entsprechendes Attribut gestzt, damit sie später nicht mehr auf der Liste der zu überprüfenden Teilnehmer auftauchen.

#### **Nachbedingung Erfolg**

Die entsprechenden Daten werden für die gewählte Veranstaltung übernommen.

#### **Nachbedingung Fehlschlag**

Es werden keine Daten für die entsprechende Veranstaltung übernommen.

#### **Interaktionen**

AKT Admin

#### 3.2.6 Daten eingeben

##### **Vorbedingung**

Der Nutzer ist als AKT Admin angemeldet. Es existiert eine veröffentlichte Ausschreibung.

##### **Beschreibung**

Der Nutzer kann aus einer Auswahl die Regatta auswählen, die er zu bearbeiten wünscht. Hierbei werden nur veröffentlichte Ausschreibungen berücksichtigt. Der Nutzer kann nun auswählen, daß er Meldedaten hinzufügen will. Hierzu existiert die Option "per Hand". Nun müssen zuerst die Kontaktinformationen für den meldenden Verein eingegeben werden. Danach können beliebig viele Meldungen zu Rennen eingegeben werden. Hierbei werden die zu Verfügung stehenden Rennen und schon eingegebene Personen zur leichteren Übernahme durch den Nutzer angezeigt. Die Personen sind auf den jeweiligen Verein begrenzt, d.h. es werden nicht alle schon eingegebenen Personen angezeigt. Zum Abschluß bekommt der Nutzer noch einmal die angegebenen Daten angezeigt und kann diese für die Regatta übernehmen. Für Aktive, deren Lizenz bei den Daten der Jugendlizenzen vorhanden sind wird ein entsprechendes Attribut gesetzt, damit sie später nicht mehr auf der Liste der zu überprüfenen Teilnehmer auftauchen.

##### **Nachbedingung Erfolg**

Die entsprechenden Daten werden für die gewählte Veranstaltung übernommen.

##### **Nachbedingung Fehlschlag**

Es werden keine Daten für die entsprechende Veranstaltung übernommen.

##### **Interaktionen**

AKT Admin

#### 3.2.7 Daten überprüfen und korrigieren

##### **Vorbedingung**

Der Nutzer ist als AKT Admin angemeldet. Es existiert eine veröffentlichte Ausschreibung, zu der schon Meldedaten eingegeben worden sind.

##### **Beschreibung**

Der Nutzer kann aus einer Auswahl die Regatta auswählen, die er zu bearbeiten wünscht. Hierbei werden nur veröffentlichte Ausschreibungen berücksichtigt. Der Nutzer kann nun auswählen, daß er Meldedaten bearbeiten will. Nun kann er sich die schon eingegebenen Daten nach verschiedenen Kriterien sortiert (meldender Verein, Rennen, Jahrgänge) anzeigen lassen. Diese Daten lassen sich nun satzweise ändern oder löschen. Bei Änderungen werden wieder Plausibilitätsprüfungen vorgenommen.

##### **Nachbedingung Erfolg**

Die geänderten Daten werden übernommen, entfernte Daten werden gelöscht.

##### **Nachbedingung Fehlschlag**

Der Nutzer wird auf die Fehler in seiner Änderung hingewiesen, die eingegebenen Daten werden nicht übernommen.

##### **Interaktionen**

AKT Admin



### 3.2.8 Rennen verlosen

#### **Vorbedingung**

Der Nutzer ist als AKT Admin angemeldet. Es existiert eine Ausschreibung, zu der schon Meldedaten eingegeben worden sind. Der Meldeschluß ist vorbei.

#### **Beschreibung**

Der Nutzer erhält anschließend eine Übersicht aller Rennen. Per Optionsfeld kann er die zu verlosenden Rennen auswählen. Mit einem Klick beginnt das System die Verlosung der ausgewählten Rennen. Dabei werden mit Hilfe von Zufallszahlen die Mannschaften der gleichen Alters- und Leistungsgruppe durchmischt. Dabei bekommt jede Mannschaft eine Zahl vergeben und nach diesen wird dann sortiert. Stellt das System fest, daß in *einer* Abteilung eines Rennen mit mehreren Abteilungen *nur* Boote *eines* Vereins stehen, zeigt das System eine Warnung an und bietet dem Nutzer die Möglichkeit das Rennen neu zu verlosen <sup>a</sup>. Vor der Verlosung überprüft das System die vorliegenden Meldedaten. Existiert für das Rennen oder eine Abteilung nur eine Meldung, oder nur Meldungen eines Vereins, so gilt das Rennen als nicht zustande gekommen. Der Nutzer bekommt dies angezeigt und hat nun die Möglichkeit die Meldungen zu löschen, in ein anderes Rennen zu verschieben oder - falls vorhanden - die Alternativmeldung zu aktivieren. Außerdem kann der Nutzer diesen Umstand akzeptieren und das Rennen trotzdem verlosen lassen. Anschließend vergibt das System über die Aufteilung in Alters- und Leistungsgruppen hinweg aufsteigende Startnummern an die Mannschaften. Unter Berücksichtigung der Alters- und Leistungsgruppen erstellt das System nun eine Unterteilung des Rennens in Abteilungen. Dieses ist insbesondere für die Rennzeitberechnung unerlässlich. Bei der Verteilung auf Abteilungen orientiert sich das System an den Bestimmungen der RWR.

#### **Nachbedingung Erfolg**

Die gewählten Rennen sind verlost worden. Die gemeldeten Mannschaften sind mit Startnummern versehen und in Abteilungen eingeteilt worden. Dabei werden die RWR-Bestimmungen eingehalten, soweit der Nutzer nicht explizit anderes vorgegeben hat. Die verlostten Rennen stehen für die Veröffentlichung des Meldeergebnisses zur Verfügung.

#### **Nachbedingung Fehlschlag**

An der Einteilung der Mannschaften hat sich nichts geändert.

#### **Interaktionen**

AKT Admin

---

<sup>a</sup>Alternativ kann das auch automatisch geschehen

#### 3.2.9 Rennzeiten festlegen

##### **Vorbedingung**

Der Nutzer ist als AKT Admin angemeldet. Es existiert eine Ausschreibung, zu der schon eine Rennverlosung stattgefunden hat.

##### **Beschreibung**

Sind die Rennen verlost, kann der Nutzer die Rennzeiten festlegen. Hierfür errechnet das System die Startzeiten. Dafür werden die in der Ausschreibung vorhandenen Werte für den Rennabstand genommen. Die Startzeit eines Rennens  $n$  errechnet sich als  $Startzeit_n = Startzeit_{n-1} + (Rennabstand * AnzahlAbteilungen_{Rennen_{n-1}})$ . Die Startzeit für das erste Rennen wird ebenfalls der Ausschreibung entnommen. Ist die Startzeitberechnung für die Rennen abgeschlossen erhält der Nutzer eine Übersicht über die Rennen, sortiert nach den Startzeiten. Er hat nun die Möglichkeit Änderungen am Zeitplan vorzunehmen. Dazu kann er: Pausen einfügen, Rennen einfügen, Rennen zusammenlegen oder die Rennen in Ihrer Reihenfolge verschieben. Nach jeder Änderung passt das System die Startzeiten an. Beim Einfügen von Rennen und dem Verschieben von Rennen werden die beteiligten Ruderer überprüft. Wird für einen der Teilnehmer der vorgeschriebene Rennabstand nicht eingehalten, so wird eine Warnung an den Nutzer ausgegeben, der diese Tatsache ignorieren oder die Änderung rückgängig machen kann. Eine Veröffentlichung des Meldeergebnisses darf *auf keinen Fall* vor der endgültigen Festlegung des Zeitplans möglich sein.

##### **Nachbedingung Erfolg**

Der vom Nutzer akzeptierte Zeitplan wird gespeichert und steht zur Veröffentlichung zur Verfügung.

##### **Nachbedingung Fehlschlag**

An der Sortierung der Rennen hat sich nichts geändert.

##### **Interaktionen**

AKT Admin

### 3.2.10 Meldeergebnis veröffentlichen

#### **Vorbedingung**

Der Nutzer ist als AKT Admin angemeldet. Es existiert eine veröffentlichte Ausschreibung, zu der schon Meldedaten eingegeben worden sind.

#### **Beschreibung**

Der Nutzer kann aus einer Auswahl die Regatta auswählen, die er zu bearbeiten wünscht. Hierbei werden nur veröffentlichte Ausschreibungen berücksichtigt. Der Nutzer kann nun auswählen, daß er ein Meldeergebnis veröffentlichen will. Dann exportiert das System die relevanten Daten in verschiedene vom Nutzer wählbare Formate. Als Exportformate werden hierbei PDF, HTML und xregatta unterstützt. Diese Daten werden auch auf dem Server gehalten und können durch den Nutzer mittels des Browsers heruntergeladen werden. Eine weitere Veränderung der Veranstaltung ist jederzeit möglich.

#### **Nachbedingung Erfolg**

Das Meldeergebnis wird in den gewählten Formaten veröffentlicht.

#### **Nachbedingung Fehlschlag**

Es werden keine Dokumente erzeugt.

#### **Interaktionen**

AKT Admin

#### 3.2.11 Nachmeldung erfassen

##### **Vorbedingung**

Der Nutzer ist als AKT Admin angemeldet. Es existiert eine veröffentlichte Ausschreibung, zu der schon Meldedaten eingegeben worden sind.

##### **Beschreibung**

Der Nutzer kann aus einer Auswahl die Regatta auswählen, die er zu bearbeiten wünscht. Hierbei werden nur veröffentlichte Ausschreibungen berücksichtigt. Der Nutzer kann nun wählen, daß er eine Nachmeldung eingeben will. Nun wird ihm die Liste der Rennen der ausgewählten Regatta präsentiert. Wird aus dieser Liste ein Rennen ausgewählt, werden die bisher gemeldeten Boote und ihre Einteilung in Läufe angezeigt. Nun kann der Nutzer die nachgemeldete Mannschaft eintragen. Dazu muß er wie bei einer normalen Meldung den Verein und die Ruderer eintragen. Hierbei werden wieder die schon eingegebenen Vereine und Personen zur Übernahme angezeigt. Ist die nachgemeldete Mannschaft eingetragen, wird sie auch in der Übersicht bei der Verteilung auf die Läufe angezeigt. Ist eine Nachmeldung nach den RWR nicht möglich (wegen des Typs der Regatta oder einer Entstehung von neuen Abteilungen), so zeigt das System einen entsprechenden Hinweis an. Um eine möglichst große Flexibilität zu gewährleisten obliegt die Entscheidung, ob eine Nachmeldung stattfinden kann trotzdem dem Nutzer. Die nachfolgende gleichmäßige Aufteilung der Läufe führt das System durch. Die nachgemeldete Mannschaft wird für das entsprechende Rennen mit dem Attribut "nachgemeldet" eingetragen, ist also in Meldeergebnissen vorhanden. Ein entsprechender Zahlungsvermerk wird erzeugt (zu der erhöhten Nachmeldegebühr) und taucht in der Funktion "Kassenbuchungen erstellen" auf.

##### **Nachbedingung Erfolg**

Die nachgemeldete Mannschaft ist in dem gewählten Rennen gemeldet, die zu startenden Abteilungen sind gleichmäßig groß. Ein entsprechender Abrechnungssatz ist erzeugt worden.

##### **Nachbedingung Fehlschlag**

Die teilnehmenden Mannschaften und die Aufteilung in Abteilungen hat sich nicht verändert. Dem Verein fallen keine zusätzlichen Gebühren an.

##### **Interaktionen**

AKT Admin

#### 3.2.12 Abmeldung erfassen

##### **Vorbedingung**

##### **Beschreibung**

TODO

##### **Nachbedingung Erfolg**

##### **Nachbedingung Fehlschlag**

##### **Interaktionen**

AKT Admin

### 3.2.13 Ummeldung erfassen

#### **Vorbedingung**

#### **Beschreibung**

TODO

#### **Nachbedingung Erfolg**

#### **Nachbedingung Fehlschlag**

#### **Interaktionen**

AKT Admin

### 3.2.14 Jugendlizenzen erfassen

#### **Vorbedingung**

Der Nutzer ist als AKT Admin angemeldet.

#### **Beschreibung**

Der Nutzer kann auswählen, daß er Jugendlizenzen eingeben möchte. Hierbei stehen ihm verschiedene Möglichkeiten zur Verfügung. Zum einen kann er eine existierende Liste in elektronischer Form übernehmen<sup>a</sup> oder von Hand eingeben. Bei der Übernahme aus der elektronischen Liste werden die enthaltenen Daten wie bei der Erfassung elektronischer Meldedaten angezeigt und können dann übernommen werden. Bei der manuellen Erfassung sind die notwendigen Daten (Verein, Ruderer, Geburtsdatum ...) vom Nutzer per Hand einzugeben. Ist die Eingabe beendet werden bei eventuell vorhandenen und noch nicht abgeschlossenen Regatten die gemeldeten Aktiven überprüft. Sind Aktive vorhanden, deren Lizenz neu eingetragen worden ist werden diese entsprechend markiert, damit sie nicht mehr in der Liste der zu überprüfenden Teilnehmer auftauchen.

#### **Nachbedingung Erfolg**

Neu übernommene Jugendlizenz Daten sind im System vorhanden und können zur Überprüfung genutzt werden.

#### **Nachbedingung Fehlschlag**

Die vorhandenen Daten werden nicht geändert.

#### **Interaktionen**

AKT Admin

---

<sup>a</sup>Hier sollte auch noch die Standardisierung eines Formates stattfinden, um den Schmerz möglichst klein zu halten. Ich weiß nicht, was die DRJ liefern kann, aber Excel pur ist vielleicht nicht so schön. Denkbar wäre, daß wir oder werow.com oder so die aktuelle Tabelle der DRJ bearbeiten und dann in einem passenden XML(?) -Format zur Verfügung stellen um den Nutzern diesen Schritt möglichst einfach zu machen.

#### 3.2.15 Jugendlizenzen kontrollieren

##### **Vorbedingung**

Der Nutzer ist als AKT Admin angemeldet. Es existiert eine veröffentlichte Ausschreibung, zu der schon Meldedaten eingegeben worden sind.

##### **Beschreibung**

Der Nutzer bekommt in einer Liste die Aktiven angezeigt, die ein Jugendlizenz vorlegen müssen. Für die jeweilige Veranstaltung (Ausschreibung) hat jeder Aktive ein Attribut JL, das anzeigt, ob eine gültige Jugendlizenz vorliegt, oder nicht. Ein automatischer Abgleich zwischen der Liste der zu kontrollierenden Teilnehmern erfolgt automatisch bei der Eingabe der Meldedaten oder von neuen Lizenzdaten. Hierbei wird auf Gleichheit von Verein, Name und Geburtsjahr getestet. Ist eine entsprechende Lizenz vorhanden, wird das Attribut auf `true` gesetzt und der jeweilige Aktive taucht nicht mehr in der Liste der zu überprüfenden Personen auf. Wird eine bestimmte Person ausgewählt (weil sie nicht in der Liste der Jugendlizenzen enthalten ist), so kann der Nutzer per Klick das JL-Attribut auf `true` setzen (z.B. bei Vorlage der JL). Hierbei werden die entsprechenden Daten des Aktiven auch in die Liste der JL Daten übernommen um bei späteren Veranstaltungen eine erneute manuelle Überprüfung zu vermeiden.

##### **Nachbedingung Erfolg**

Aktive, bei denen die Existenz der JL manuell bestätigt wurde werden nicht mehr in der Liste angezeigt. Ihre Daten werden in die JL-Liste übernommen.

##### **Nachbedingung Fehlschlag**

Die Liste der zu überprüfenden Aktiven ändert sich nicht.

##### **Interaktionen**

AKT Admin

#### 3.2.16 Schiedsrichtereinsatzplan erstellen

##### **Vorbedingung**

##### **Beschreibung**

TODO

##### **Nachbedingung Erfolg**

##### **Nachbedingung Fehlschlag**

##### **Interaktionen**

AKT Admin

### 3.2.17 Schiedsrichtereinsatzplan veröffentlichen

**Vorbedingung**

**Beschreibung**

TODO

**Nachbedingung Erfolg**

**Nachbedingung Fehlschlag**

**Interaktionen**

AKT Admin

### 3.2.18 Waage erfassen

**Vorbedingung**

Der Nutzer ist als AKT Admin oder AKT Waage angemeldet. Es existieren Meldungen zu LGW Rennen.

**Beschreibung**

Der Nutzer bekommt alle Meldungen zu LGW-Rennen bzw. Steuerleute nach Vereinen getrennt angezeigt. Wählt er eine bestimmte Person aus, werden außerdem die in Frage kommenden Rennen (Renn-Nr., Bezeichnung, Startzeit) angezeigt. Der Nutzer kann nun das bestimmte Gewicht des Aktiven eintragen. Den Check zu den Gewichtsgrenzen führt das System selbständig durch, da hier verschiedenen Grenzen für denselben Aktiven gelten können.

**Nachbedingung Erfolg**

Das Gewicht wird für den jeweiligen Aktiven gespeichert.

**Nachbedingung Fehlschlag**

Es werden keine Daten gespeichert.

**Interaktionen**

AKT Waage, AKT Admin

#### 3.2.19 Kassenbuchungen erstellen

##### **Vorbedingung**

Der Nutzer ist als AKT Kasse oder AKT Admin angemeldet. Es existiert eine veröffentlichte Ausschreibung, zu der schon Meldedaaten eingegeben worden sind.

##### **Beschreibung**

Der Nutzer bekommt alle meldenden Verein und Renngemeinschaften sowie deren fällige Startgelder angezeigt. Wählt er einen bestimmten Verein aus der Liste aus bekommt alle Rennen und deren Kosten im einzelnen angezeigt. Hier kann der Nutzer nun einzelne oder alle Rennen als bezahlt markieren. Die zu leistende Summe des Vereins wird um die entsprechende Summe reduziert. Fällt die Summe des fälligen Startgeldes auf 0,- EUR wird der Verein nicht mehr in der Liste angezeigt. Das System führt auch Nachmeldegelder und Pfand für Einsprüche mit auf, so daß ein Verein, der bereits aus der Liste verschwunden ist wieder dort auftauchen kann, wenn noch weitere Gelder fällig werden, z.B. durch Nachmeldungen.

##### **Nachbedingung Erfolg**

Die bezahlten Rennen und Beträge werden für die meldenden Vereine gespeichert.

##### **Nachbedingung Fehlschlag**

Es werden keine Daten gespeichert.

##### **Interaktionen**

AKT Kasse, AKT Admin

#### 3.2.20 Quittung erstellen

##### **Vorbedingung**

Der Nutzer ist als AKT Kasse oder AKT Admin angemeldet. Es sind schon Buchungen vorgenommen worden.

##### **Beschreibung**

Der Nutzer bekommt alle Vereine für die schon Buchungen vorgenommen worden sind angezeigt. Wählt er einen bestimmten Verein aus der Liste aus, so bekommt er die bereits bezahlten Rennen angezeigt. Hierbei werden die Rennnr., die Rennbezeichnung, die Anzahl der Meldungen in dem Rennen, die Einzelkosten pro Meldung und die akkumulierten Kosten für dieses Rennen aufgeführt. Des weiteren enthält die Ansicht den Namen der Veranstaltung und des jeweiligen Vereins, sowie eine Gesamtsumme. Diese Ansicht läßt sich über die Druckfunktion des Browsers drucken und dient dann als Quittung.

##### **Nachbedingung Erfolg**

Die Quittung für den jeweiligen Verein und die bezahlten Rennen wird angezeigt.

##### **Nachbedingung Fehlschlag**

Es wird keine Quittung für den Verein angezeigt.

##### **Interaktionen**

AKT Kasse, AKT Admin



### 3.2.21 Berichte erstellen

**Vorbedingung**

**Beschreibung**

TODO

**Nachbedingung** Erfolg

**Nachbedingung** Fehlschlag

**Interaktionen**

AKT Kasse, AKT Admin

### 3.2.22 Rennen verlegen

**Vorbedingung**

**Beschreibung**

TODO

**Nachbedingung** Erfolg

**Nachbedingung** Fehlschlag

**Interaktionen**

AKT Admin

### 3.2.23 Einsprüche erfassen

**Vorbedingung**

**Beschreibung**

TODO

**Nachbedingung** Erfolg

**Nachbedingung** Fehlschlag

**Interaktionen**

AKT Admin

### 3.2.24 Schiedsrichterbericht erstellen

**Vorbedingung**

**Beschreibung**

TODO

**Nachbedingung** Erfolg

**Nachbedingung** Fehlschlag

**Interaktionen**

AKT Admin

### 3.2.25 Urkunden drucken

#### Vorbedingung

Der Nutzer ist als AKT Admin angemeldet. Es existieren Rennen in denen für alle Starter Zeiten eingetragen sind.

#### Beschreibung

Der Nutzer bekommt eine Liste aller abgeschlossenen Rennen angezeigt. Das sind die Rennen, in denen für alle Starter ein Ergebnis (entweder eine Zeit oder abgemeldet oder gekentert) vermerkt ist. Wählt der Nutzer ein bestimmtes Rennen aus, bekommt er die Rangfolge der gestarteten Mannschaften angezeigt. Diese wird vom System erzeugt. Die Sortierung entspricht den Regulierungen des DRV<sup>a</sup>. Aus dieser Liste kann der Nutzer nun Mannschaften auswählen, für die Urkunden gedruckt werden sollen. Des weiteren kann der Nutzer ein Urkunden-Template auswählen, in das die Urkunden gedruckt werden sollen. Als Vorlage dienen hierbei Templates, in die bestimmte Token eingebettet werden können (#NAME, #ZEIT, #PLATZ usw.). Diese werden vom System durch die jeweiligen Daten der Mannschaft ersetzt<sup>b</sup>. Berücksichtigt werden alle Dateien in einem bestimmten Template-Verzeichnis, die möglichen Token werden von **ROWORE** vorgegeben. Zielformat hierbei ist PDF wegen der gestalterischen Vorteile, die es hinsichtlich der absoluten Seitengestaltung bietet.<sup>c</sup>

*EXPERIMENTAL:* Um das Drucken von Urkunden zu erleichtern sollte das ganze so konzipiert werden, daß man dem System eine Regeldatei liefern kann, die vorgibt was für Urkunden zu welchen Rennen gedruckt werden sollen. Auch diese Regeldateien sollten sich leicht anlegen lassen um den verschiedenen Wünschen verschiedener Nutzer entgegen zu kommen.

#### Nachbedingung Erfolg

Die Urkunden werden mit den Daten der Mannschaften des gewählten Rennens in das gewählte Urkunden Template gedruckt.

#### Nachbedingung Fehlschlag

Es werden keine Daten gedruckt.

#### Interaktionen

AKT Admin

---

<sup>a</sup>Demian hatte auf ein entsprechendes Dokument hingewiesen. Dies sollte einfach als Teil dieser Spezifikation mit eingebunden werden und kann dann an dieser Stelle referenziert werden.

<sup>b</sup>Das Format dieser Templates ist noch unklar, sollte aber möglichst einfach sein um auch breite Anwendung zu finden.

<sup>c</sup>XML FO ist ein erster interessanter Technologie-Kandidat

#### 3.2.26 Ergebnisse veröffentlichen

##### **Vorbedingung**

Der Nutzer ist als AKT Admin angemeldet. Alle Rennen der gewählten Veranstaltung sind abgeschlossen.

##### **Beschreibung**

Sind alle Rennen einer Veranstaltung abgeschlossen, d.h. für alle Starter ist ein Ergebnis (entweder eine Zeit oder abgemeldet oder gekentert) vermerkt oder ein Rennen ist als 'ausgefallen' markiert, kann der Nutzer das Ergebnis der Veranstaltung veröffentlichen. Hierbei wird nach Rennen sortiert die Rangfolge der gestarteten Mannschaften aufgeführt. Diese wird vom System erzeugt. Die Sortierung entspricht den Regulierungen des DRV<sup>a</sup>. Des Weiteren werden der Name und das Datum der Veranstaltung aufgeführt. Dann exportiert das System die relevanten Daten in verschiedene vom Nutzer wählbare Formate. Als Exportformate werden hierbei PDF, HTML und xregatta unterstützt. Diese Daten werden auch auf dem Server gehalten und können durch den Nutzer mittels des Browsers heruntergeladen werden.

##### **Nachbedingung Erfolg**

Das Ergebnis wird in den gewählten Formaten veröffentlicht.

##### **Nachbedingung Fehlschlag**

Es werden keine Daten veröffentlicht.

##### **Interaktionen**

AKT Admin

---

<sup>a</sup>Demian hatte auf ein entsprechendes Dokument hingewiesen. Dies sollte einfach als Teil dieser Spezifikation mit eingebunden werden und kann dann an dieser Stelle referenziert werden.

#### 3.2.27 Startberechtigung kontrollieren

##### **Vorbedingung**

##### **Beschreibung**

TODO

##### **Nachbedingung Erfolg**

##### **Nachbedingung Fehlschlag**

##### **Interaktionen**

AKT Start, AKT Admin

### 3.2.28 Ausschluss von Mannschaften erfassen

**Vorbedingung**

**Beschreibung**

TODO

**Nachbedingung Erfolg**

**Nachbedingung Fehlschlag**

**Interaktionen**

AKT Start, AKT Ziel

### 3.2.29 Mannschaften zu Rennen einteilen

**Vorbedingung**

**Beschreibung**

TODO

**Nachbedingung Erfolg**

**Nachbedingung Fehlschlag**

**Interaktionen**

AKT Start, AKT Admin

### 3.2.30 Rennen starten

**Vorbedingung**

**Beschreibung**

TODO

**Nachbedingung Erfolg**

**Nachbedingung Fehlschlag**

**Interaktionen**

AKT Start, AKT Admin

### 3.2.31 Zwischenzeiten nehmen

#### **Vorbedingung**

#### **Beschreibung**

TODO

**Nachbedingung** Erfolg

**Nachbedingung** Fehlschlag

#### **Interaktionen**

AKT Zeit, AKT Admin

### 3.2.32 Unfälle erfassen

#### **Vorbedingung**

#### **Beschreibung**

TODO

**Nachbedingung** Erfolg

**Nachbedingung** Fehlschlag

#### **Interaktionen**

AKT Zeit, AKT Start, AKT Ziel, AKT Admin

### 3.2.33 Einsprüche erfassen

#### **Vorbedingung**

#### **Beschreibung**

TODO

**Nachbedingung** Erfolg

**Nachbedingung** Fehlschlag

#### **Interaktionen**

AKT Admin

### 3.2.34 Zeit nehmen

#### Vorbedingung

#### Beschreibung

TODO

#### Nachbedingung Erfolg

#### Nachbedingung Fehlschlag

#### Interaktionen

AKT Ziel, AKT Admin

### 3.2.35 Zielfotos hochladen

#### Vorbedingung

#### Beschreibung

TODO

#### Nachbedingung Erfolg

#### Nachbedingung Fehlschlag

#### Interaktionen

AKT Ziel, AKT Admin

### 3.2.36 Rennzeiten einsehen

#### Vorbedingung

Der Nutzer ist als AKT Gast angemeldet. Es existieren Rennen, die schon beendet sind.

#### Beschreibung

Besucher der Veransaltung können Zeiten von abgeschlossenen Rennen einsehen. Hierfür wird dem Nutzer eine Liste der schon abgeschlossenen Rennen angezeigt. Das sind die Rennen, in denen für alle Starter ein Ergebnis (entweder eine Zeit oder abgemeldet oder gekentert) vermerkt ist. Wählt der Nutzer ein bestimmtes Rennen aus, bekommt er die Rangfolge der gestarteten Mannschaften angezeigt. Diese wird vom System erzeugt. Die Sortierung entspricht den Regulierungen des DRV<sup>a</sup>.

#### Nachbedingung Erfolg

Die gewählten Daten werden angezeigt.

#### Nachbedingung Fehlschlag

Keine

#### Interaktionen

AKT Gast

---

<sup>a</sup>Demian hatte auf ein entsprechendes Dokument hingewiesen. Dies sollte einfach als Teil dieser Spezifikation mit eingebunden werden und kann dann an dieser Stelle referenziert werden.

#### 3.2.37 Neue Nutzer anlegen

##### **Vorbedingung**

Der Nutzer ist als AKT Sysadmin angemeldet.

##### **Beschreibung**

Um den Nutzern des Systems einen eigenen Login zur Verfügung zu stellen soll **ROW@RE** Nutzer verwalten. Diese werden vom AKT Sysadmin angelegt. Hierbei wird der Nutzernamen, das Passwort und die Zugehörigkeit zu den möglichen Rollen im System eingetragen. Systemseitig existieren zwei Accounts fest, das sind "Gast" und "Sysadmin". Der Nutzer kann nun in einer Maske den Nutzernamen, das Passwort und die möglichen Gruppen des neuen Nutzers eintragen. Dabei ist eine Zugehörigkeit zu beliebig vielen Gruppen möglich.

##### **Nachbedingung Erfolg**

Der neu angelegte Nutzer ist im System gespeichert und kann sich anmelden.

##### **Nachbedingung Fehlschlag**

Es wird kein neuer Nutzer angelegt.

##### **Interaktionen**

AKT Sysadmin

#### 3.2.38 Nutzer bearbeiten

##### **Vorbedingung**

Der Nutzer ist als AKT Sysadmin angemeldet. Es existieren Accounts für das System.

##### **Beschreibung**

Um Nutzer verwalten zu können müssen die Daten geändert werden können. Der Nutzer bekommt eine Liste der im System vorhandenen Nutzer angezeigt. Wählt er einen aus, bekommt er die Daten dieses Nutzers angezeigt. Diese können nun geändert und im System gespeichert werden.

##### **Nachbedingung Erfolg**

Die geänderten Daten sind im System übernommen.

##### **Nachbedingung Fehlschlag**

Die Daten des Nutzers werden nicht geändert. <sup>a</sup>

##### **Interaktionen**

AKT Sysadmin

---

<sup>a</sup>Ich denke dies sollte z.B. passieren, wenn der geänderte Nutzer gerade angemeldet ist, das erspart ein paar diese Zustandsüberprüfungen



### 3.2.39 Nutzer löschen

#### **Vorbedingung**

Der Nutzer ist als AKT Sysadmin angemeldet. Es existieren Accounts für das System.

#### **Beschreibung**

Der Nutzer bekommt eine Liste der im System vorhandenen Nutzer angezeigt. Wählt er einen bestimmten aus, bekommt er die Daten angezeigt und kann diesen nach einer Bestätigung "Wollen Sie das wirklich tun?" löschen.

#### **Nachbedingung Erfolg**

Der gelöschte Nutzer ist nicht mehr im System vorhanden und kann sich mit seinen Daten nicht mehr anmelden.

#### **Nachbedingung Fehlschlag**

Die Daten des Nutzers werden nicht gelöscht. <sup>a</sup>

#### **Interaktionen**

AKT Sysadmin

---

<sup>a</sup>wie darüber.

DRAFT

## 4 Nichtfunktionale Anforderungen

### 4.1 Leistungsanforderungen

TODO: Mengengerüst, Kommunikation, Synchronisation der Daten

### 4.2 Entwurfseinschränkungen

TODO: Nur PHP/PostgreSQL; HTML, JavaScript ?

### 4.3 Verfügbarkeit

Keine besonderen Anforderungen bzgl. Verfügbarkeit.

### 4.4 Sicherheit

TODO: Daten werden in Datenbank persistent gespeichert, regelmässige automatische Backups etc.

### 4.5 Robustheit

TODO: Robustheit gegenüber fehlerhaften Einträgen, Validierung etc

### 4.6 Wartbarkeit

TODO: Entwurfsregeln zur Verbesserung der Wartbarkeit des PHPcodes, z.B. Objektorientierung, Daten- kapselung und -abstraktion, Trennung von Businesslogik und Darstellungsschicht.

### 4.7 Portabilität

TODO: Windows, Unix-derivate -> (W|L)AMP

DRAFT

## 5 Graphische Benutzeroberfläche

TODO

### 5.1 Anmeldefenster

TODO

### 5.2 Hauptfenster

TODO

DRAFT

DRAFT

## 6 Daten

TODO Datenstruktur beschreiben

DRAFT

DRAFT



## 7 Begriffslexikon

DRAFT

DRAFT

## 8 Offene Fragen

DRAFT