

# KANU

Kanu s.a.M.



Ein Beitrag zum **Inverso** Softwarewettbewerb in  
der Kategorie **Schüler**

# s.a.M.

# Inhalt

- 1 Projektteilnehmer
- 2 Projektbeschreibung
- 3 Funktionsumfang und Einsatzszenarien
- 4 Entwicklungsablauf, eigene Leistungen
- 5 Technische Spezifikation
- 6 Ergebnisse
- 7 Originalität

# Projektteilnehmer

# 1 Projektteilnehmer



- ▼ Name: Eric Ackermann
- ▼ Schule: Carl-Zeiss-Gymnasium mit math.-nat-techn. Spezialklassen Jena
- ▼ Klassenstufe: 12
- ▼ Erfahrung im Bereich Informatik:
  - ▼ Leistungskurs Informatik, 6 Wochenstunden
  - ▼ Dreimalige Teilnahme am Bundeswettbewerb Informatik
  - ▼ mit Sonderpreis im Landeswettbewerb Jugend forscht ausgezeichnete Trainingssoftware für Sportschützen

# Projektbeschreibung

## 2 Projektbeschreibung

- ▼ Notwendigkeit der Erhebung und Zuordnung von Zeiten im Kanuslalom in Training und Wettkampf: Laufzeiten und Strafzeiten
- ▼ auch bei regionalen Vereinen wie dem SV Schott oder dem Jenaer Kanu- und Ruderverein nötig
- ▼ Hier: Streckenlänge von 200 m der Jenaer Saale, auf die 18-25 Messtore verteilt werden
- ▼ Messtore sind 8 Messstationen zugeteilt, in denen je ein Helfer den Startern Strafzeiten für falsches Passieren zuweist, wie in den Wettkampfregeln vorgeschrieben



## 2 Projektbeschreibung



**Abbildung:** Sportler bei der Durchfahrt durch ein Messtor,  
Quelle: [http://rio.sportschau.de/rio2016/kanu426\\_v-ardgalerie.jpg](http://rio.sportschau.de/rio2016/kanu426_v-ardgalerie.jpg). Version: 08.01.2017 14:47

## 2 Projektbeschreibung

- ▼ Notwendigkeit der Erhebung und Zuordnung von Zeiten im Kanuslalom in Training und Wettkampf: Laufzeiten und Strafzeiten
- ▼ auch bei regionalen Vereinen wie dem SV Schott oder dem Jenaer Kanu- und Ruderverein nötig
- ▼ Hier: Streckenlänge von 200 m der Jenaer Saale, auf die 18-25 Messtore verteilt werden
- ▼ Messtore sind 8 Messstationen zugeteilt, in denen je ein Helfer den Startern Strafzeiten für falsches Passieren zuweist, wie in den Wettkampfbregeln vorgeschrieben
- ▼ bisher: Erfassung per Hand, Auswertung mittels Papierlisten und Tabellenkalkulationen
- ▼ Zuordnung bei bis zu 27 Einzelrennen mit mindestens 2 Läufen und vielen Startern zu genau einem Sportler nötig
- ▼ → hoher Aufwand, Fehleranfälligkeit



## 2 Projektbeschreibung

- ▼ Existenz von kommerziellen Systemen zur automatischen Zeiterfassung, aber: teuer (mind. 12.000. €), Feature der Strafzeitsynchronisation fehlt
- ▼ Grund: für andere Art des Sports entwickelt, bei der Starter parallel starten und eine gerade Strecke schnell bewältigen müssen

## 2 Projektbeschreibung



**Abbildung:** Fotofinish eines kommerziellen Zeitenmesssystems, Quelle:  
<http://www.imas-sport.com/index.php/de/zeitmessung/basissystem>. Version: 26.12.2016 15:53

## 2 Projektbeschreibung

- ▼ Existenz von kommerziellen Systemen zur automatischen Zeiterfassung, aber: teuer (mind. 12.000. €), Feature der Strafzeitensynchronisation fehlt
- ▼ Grund: für andere Art des Sports entwickelt, bei der Starter parallel starten und eine gerade Strecke schnell bewältigen müssen
- ▼ zu entwickelndes Softwaresystem soll das Problem der Starterverwaltung, Wettkampfzeiterhebung, Strafzeitensynchronisation und Wettkampfauswertung beim Kanuslalom möglichst günstig und einfach lösen
- ▼ verwendete Hardware: nur Android-Smartphones und ein Windows-Rechner
- ▼ verwendete Software: nur gratis-Programme
- ▼ Verbindung der Geräte in einem lokalen Netzwerk, aber kein Internetzugriff (Manipulierbarkeit)

# Funktionsumfang und Einsatzszenarien

# 3 Funktionsumfang und Einsatzszenarien

- ▼ Produkt: Softwaresystem **Kanu s.a.M.** (semi-automatisches Messsystem)
- ▼ in enger Zusammenarbeit mit dem SV Schott, vertreten durch Herrn Schade, entwickelt

# 3 Funktionsumfang und Einsatzszenarien

- ▼ Produkt: Softwaresystem **Kanu s.a.M.** (semi-automatisches Messsystem)
- ▼ in enger Zusammenarbeit mit dem SV Schott, vertreten durch Herrn Schade, entwickelt
- ▼ Bestandteile:
  - ▼ JavaFX-basierendes Desktopprogramm “Kanu s.a.M.”, läuft auf zentralem Laptop, übernimmt Starterverwaltung, Zuordnung der Zeiten und Auswertung



# 3 Funktionsumfang und Einsatzszenarien

- ▼ Produkt: Softwaresystem **Kanu s.a.M.** (semi-automatisches Messsystem)
- ▼ in enger Zusammenarbeit mit dem SV Schott, vertreten durch Herrn Schade, entwickelt
- ▼ Bestandteile:
  - ▼ JavaFX-basierendes Desktopprogramm “Kanu s.a.M.”, läuft auf zentralem Laptop, übernimmt Starterverwaltung, Zuordnung der Zeiten und Auswertung
  - ▼ lokaler PHP- und MySQL-Server, dient der Kommunikation mit den Smartphones und der App, nicht vom Internet aus beeinflussbar

# 3 Funktionsumfang und Einsatzszenarien

- ▼ Produkt: Softwaresystem **Kanu s.a.M.** (semi-automatisches Messsystem)
- ▼ in enger Zusammenarbeit mit dem SV Schott, vertreten durch Herrn Schade, entwickelt
- ▼ Bestandteile:
  - ▼ JavaFX-basierendes Desktopprogramm “Kanu s.a.M.”, läuft auf zentralem Laptop, übernimmt Starterverwaltung, Zuordnung der Zeiten und Auswertung
  - ▼ lokaler PHP- und MySQL-Server, dient der Kommunikation mit den Smartphones und der App, nicht vom Internet aus beeinflussbar
  - ▼ Android-App “Kanu s.a.M. App”, dient dem Auslösen des Start- und Stopp- Signals und der Eintragung von Strafzeiten

# 3 Funktionsumfang und Einsatzszenarien

- ▼ Produkt: Softwaresystem **Kanu s.a.M.** (semi-automatisches Messsystem)
- ▼ in enger Zusammenarbeit mit dem SV Schott, vertreten durch Herrn Schade, entwickelt
- ▼ Bestandteile:
  - ▼ JavaFX-basierendes Desktopprogramm “Kanu s.a.M.”, läuft auf zentralem Laptop, übernimmt Starterverwaltung, Zuordnung der Zeiten und Auswertung
  - ▼ lokaler PHP- und MySQL-Server, dient der Kommunikation mit den Smartphones und der App, nicht vom Internet aus beeinflussbar
  - ▼ Android-App “Kanu s.a.M. App”, dient dem Auslösen des Start- und Stopp- Signals und der Eintragung von Strafzeiten
- ▼ System ab Android 2.3.3 und Windows XP lauffähig, an Hardware nur 1 Laptop und ein Smartphone pro Helfer nötig (ggf. weitere Hardware erforderlich zum Aufbauen eines ausreichend stabilen WLAN-Netzes)

# 3 Funktionsumfang und Einsatzszenarien

## Features des Hauptprogramms

Einlesen von Starterdaten aus Excel-Dateien

Anlegen und Initialisieren der nötigen Infrastruktur (Software)

Erstellung von Sicherungsprotokollen, die Wiederherstellung nach Absturz ermöglichen

Zugriff auf MySQL-Datenbank, um Werte mit Smartphones zu synchronisieren

Verwaltung der Starter, Zuordnung zu Kategorien, Start dieser

Verwaltung der Daten der einzelnen Läufe, Start und Stopp dieser

Anzeige der aktuellen Laufaten der jeweiligen Starter im angegebenen Intervall

Auswertung der Rennen

# 3 Funktionsumfang und Einsatzszenarien

## Features des Hauptprogramms

Einlesen von Starterdaten aus Excel-Dateien

Anlegen und Initialisieren der nötigen Infrastruktur (Software)

Erstellung von Sicherungsprotokollen, die Wiederherstellung nach Absturz ermöglichen

Zugriff auf MySQL-Datenbank, um Werte mit Smartphones zu synchronisieren

Verwaltung der Starter, Zuordnung zu Kategorien, Start dieser

Verwaltung der Daten der einzelnen Läufe, Start und Stopp dieser

Anzeige der aktuellen Laufaten der jeweiligen Starter im angegebenen Intervall

Auswertung der Rennen

## Features der App

indirekter Verbindungsaufbau zur Datenbank über PHP-Server

Anzeige des aktuellen Laufs

Handstarten eines ausgewählten Starters

Handstoppen eines ausgewählten Starters

Anzeigen der vergangenen Zeit seit Wettbewerbsstart

Auswahl der aktuellen Startnummer und ihrer Strafen an den Toren der Messstation

Eintragen der gewählten Strafzeiten an den gewählten Toren für die gewählte Startnummer

# 3 Funktionsumfang und Einsatzszenarien

Programmstart auf dem Desktoprechner



# 3 Funktionsumfang und Einsatzszenarien

Programmstart auf dem Desktoprechner



Konfiguration des Wettkampfes

# 3 Funktionsumfang und Einsatzszenarien

Startnummer	Name	Kategorie
1	Max Mustermann	Elite
3	Franz Schuster	Elite
15	Karl Schmidt	Pro
27	Franz Franz	Ober-Elite
35	Herrmann Mann	Pro

aus Datei lesen   neue Zeile   Zeile entfernen   Startnummern automatisch generieren

Bitte geben Sie hier die Zahl der Messstationen ein!

Bitte geben Sie hier die Zahl der Messtore ein!

Bitte geben Sie hier die Konfigurationswerte des MySQL-Servers ein, sofern Sie diese verändert haben!

Hostname:  Port:

Datenbank:  ☒ Datenbank automatisch leeren?

Benutzername:  Passwort:

Speicherort von XAMPP:

☒ Netzwerkverbindung vor Programmstart prüfen?

☒ Sicherungsprotokolle aktivieren?

Speicherort:

Abbildung: Konfigurationsfenster des Hauptprogramms

# 3 Funktionsumfang und Einsatzszenarien

Programmstart auf dem Desktoprechner

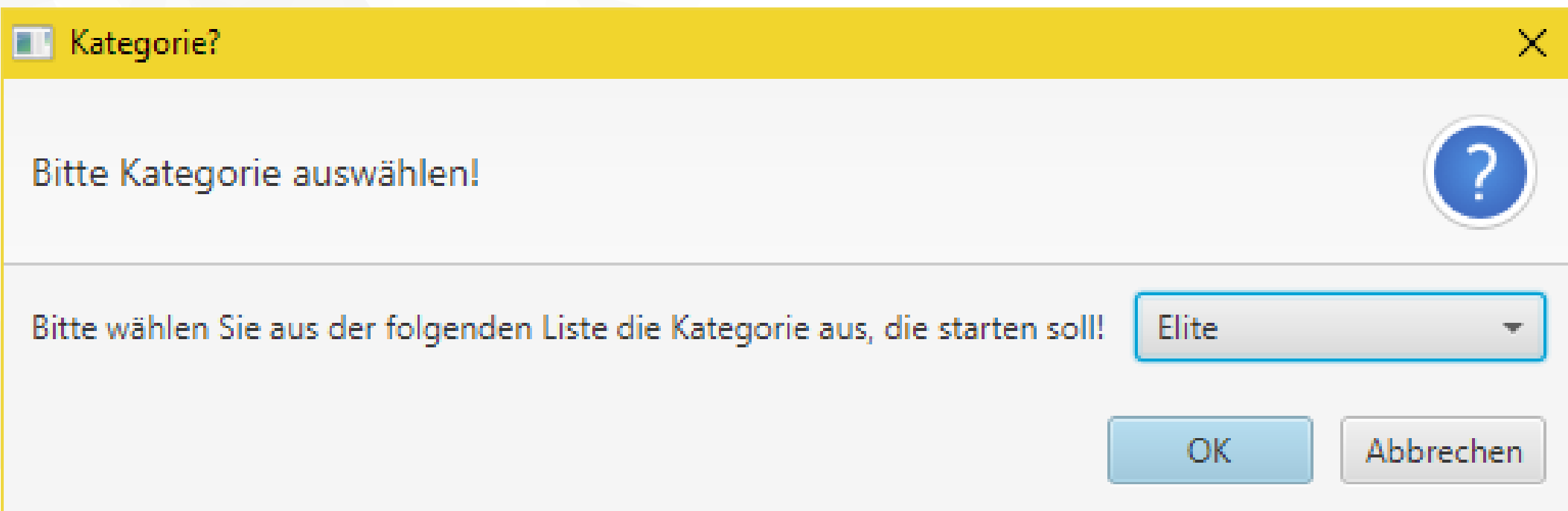


Konfiguration des Wettkampfes



Wahl der Kategorie

# 3 Funktionsumfang und Einsatzszenarien



Kategorie?

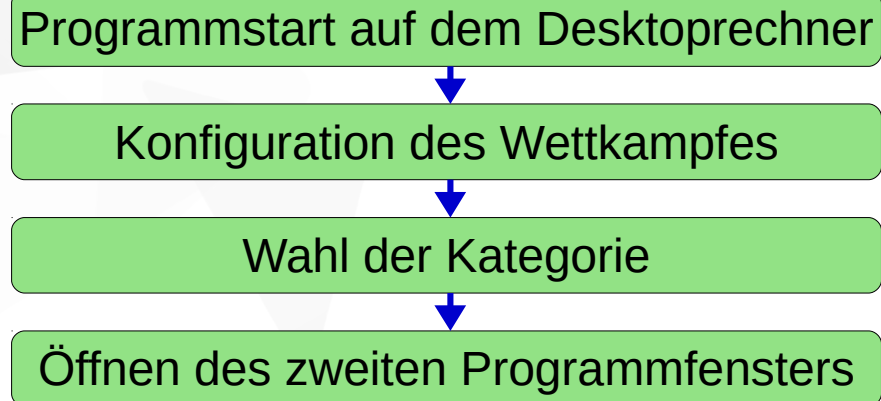
Bitte Kategorie auswählen!

Bitte wählen Sie aus der folgenden Liste die Kategorie aus, die starten soll!

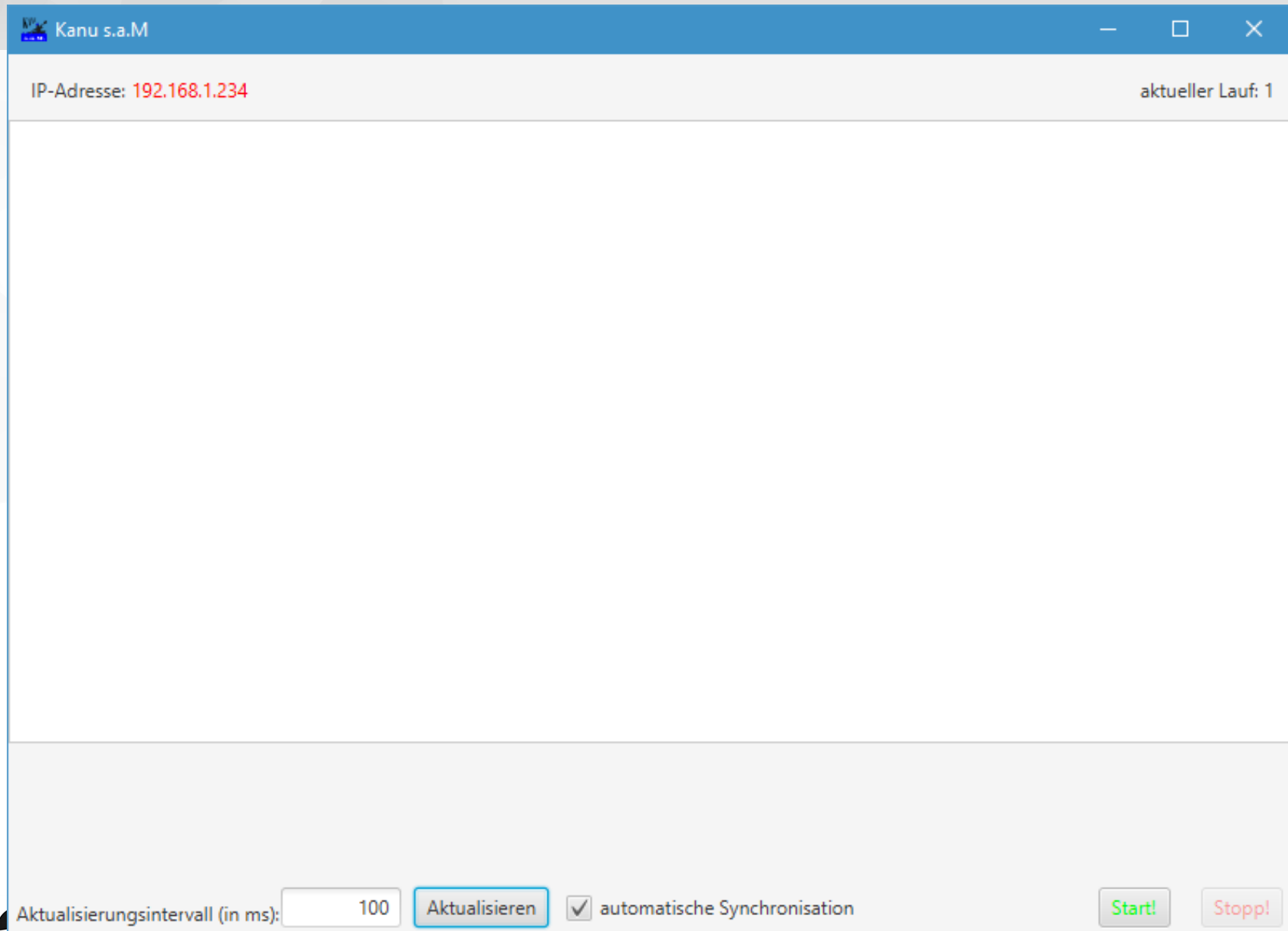
Elite

OK Abbrechen

# 3 Funktionsumfang und Einsatzszenarien



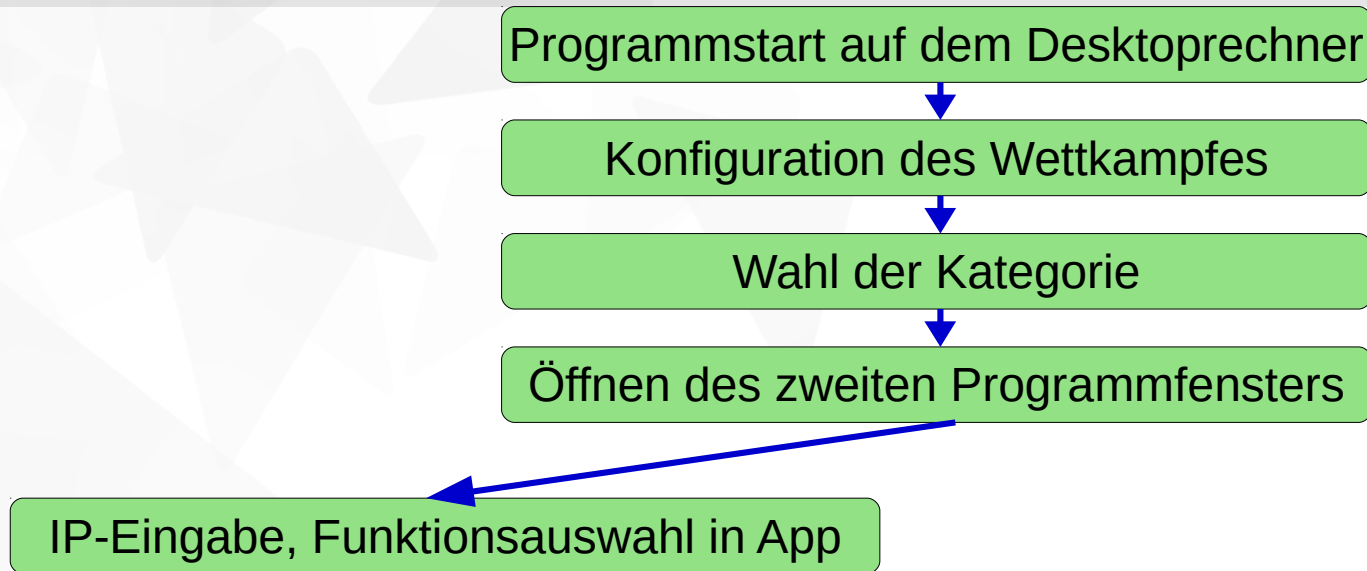
# 3 Funktionsumfang und Einsatzszenarien



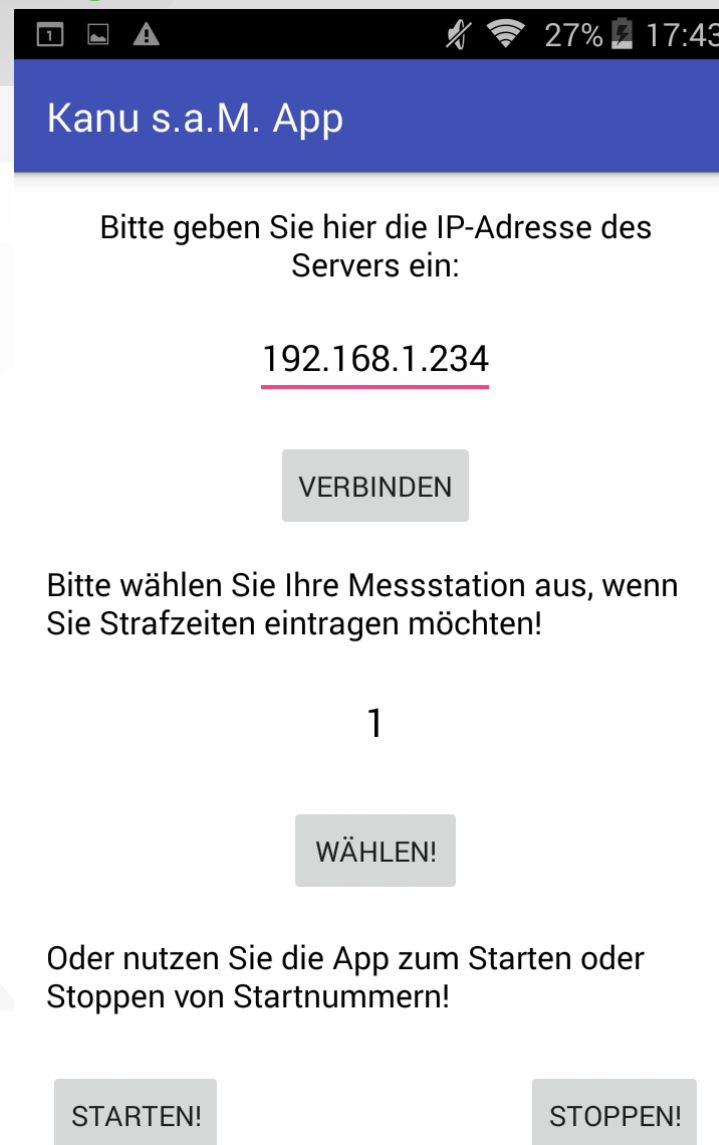
**Abbildung:** Anzeige des Hauptprogramms beim ersten Öffnen



# 3 Funktionsumfang und Einsatzszenarien



# 3 Funktionsumfang und Einsatzszenarien



The screenshot shows the 'Kanu s.a.M. App' interface on an Android device. The status bar at the top indicates 27% battery and the time 17:43. The app title 'Kanu s.a.M. App' is displayed in a blue header. The main screen prompts the user to enter the IP address of the server. The IP address '192.168.1.234' is entered and underlined in pink. Below the input field is a 'VERBINDEN' button. Further down, the app prompts the user to select a measurement station, with the number '1' displayed. A 'WÄHLEN!' button is located below the selection. At the bottom, there are two buttons: 'STARTEN!' and 'STOPPEN!'. The background of the app interface features a faint, stylized map of Germany.

Kanu s.a.M. App

Bitte geben Sie hier die IP-Adresse des Servers ein:

192.168.1.234

VERBINDEN

Bitte wählen Sie Ihre Messstation aus, wenn Sie Strafzeiten eintragen möchten!

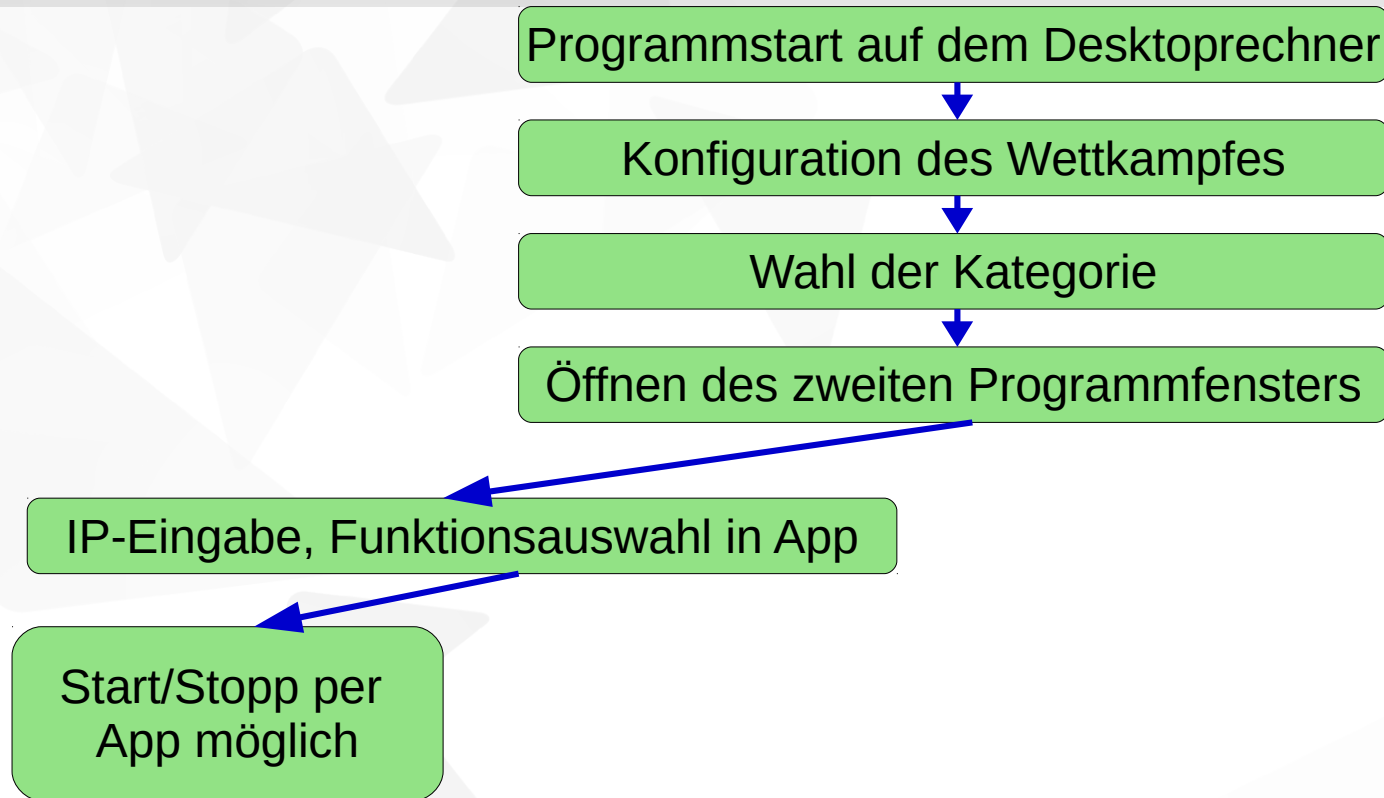
1

WÄHLEN!

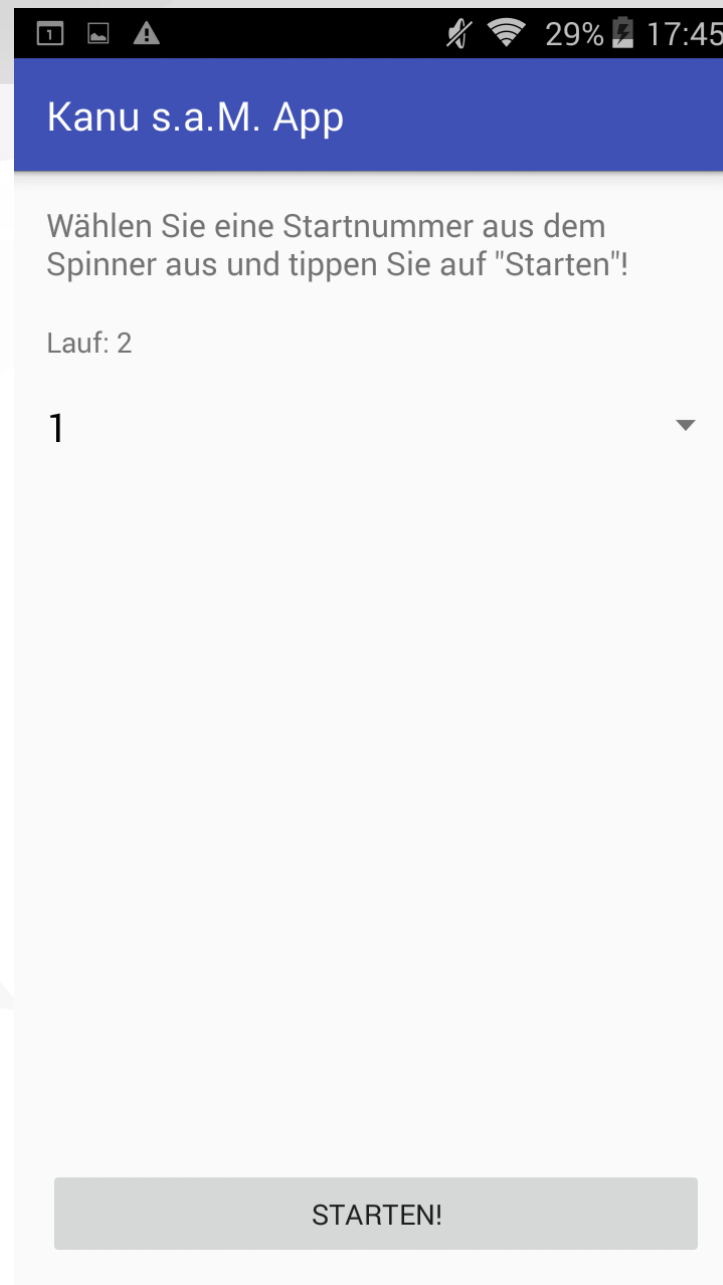
Oder nutzen Sie die App zum Starten oder Stoppen von Startnummern!

STARTEN! STOPPEN!

# 3 Funktionsumfang und Einsatzszenarien

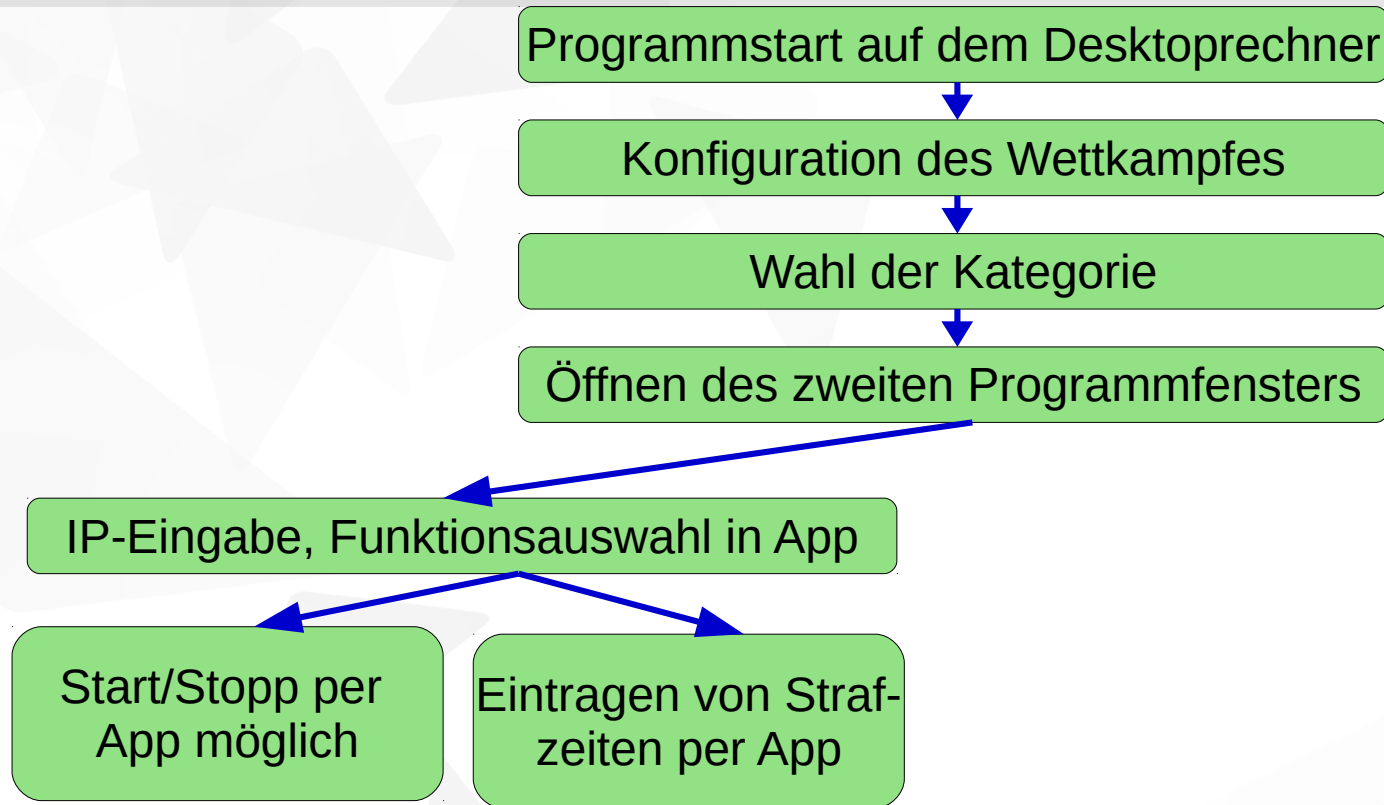


# 3 Funktionsumfang und Einsatzszenarien



**Abbildung:** Android-App in der Anzeige zum Starten

# 3 Funktionsumfang und Einsatzszenarien



# 3 Funktionsumfang und Einsatzszenarien

Kanu s.a.M. App

Eingeloggt als Messstation: 1.

Startnummer: 3▼

Tor: Strafsekunden:

1	2
4	0
7	2

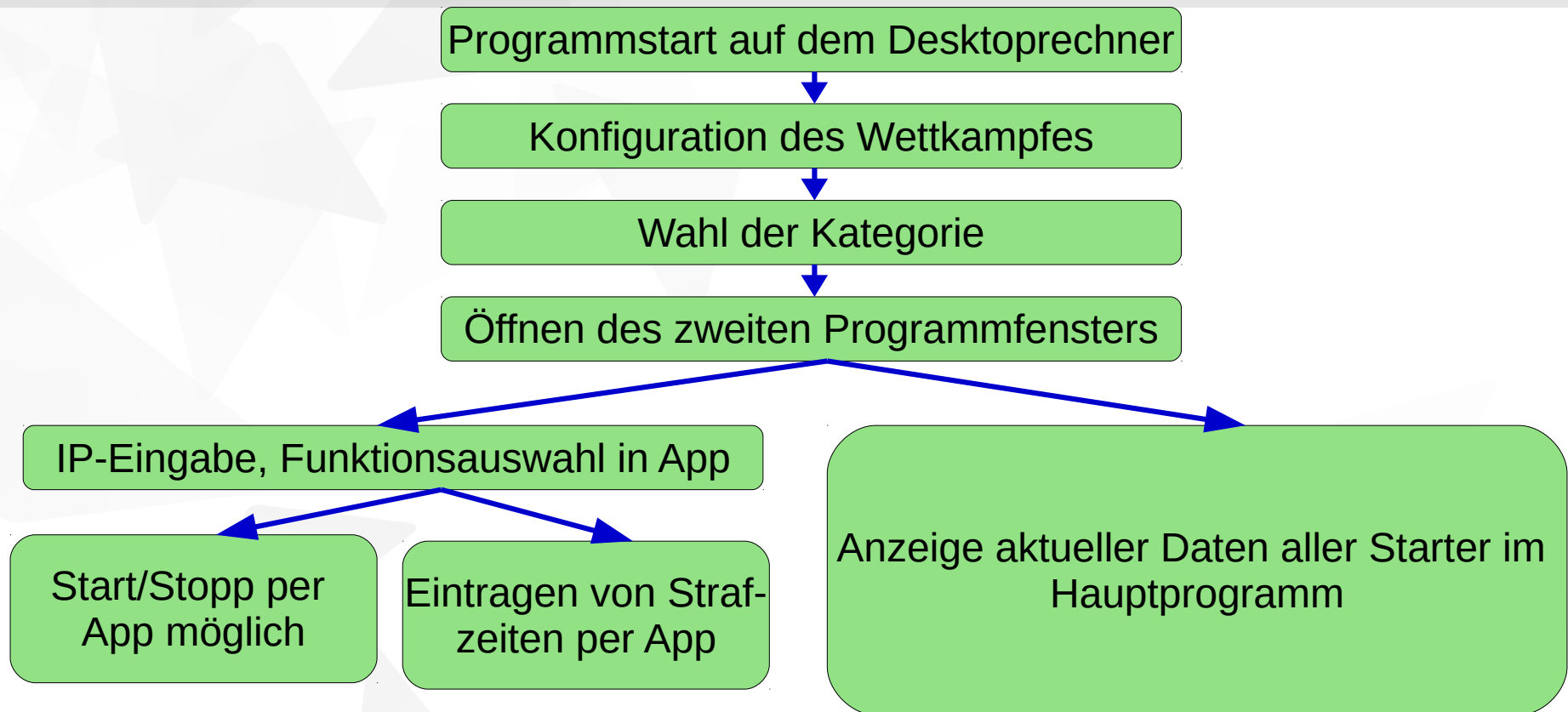
Lauf: 1

Zeit seit Start: 4 Minuten, 47 Sekunden, 479 Millisekunden.

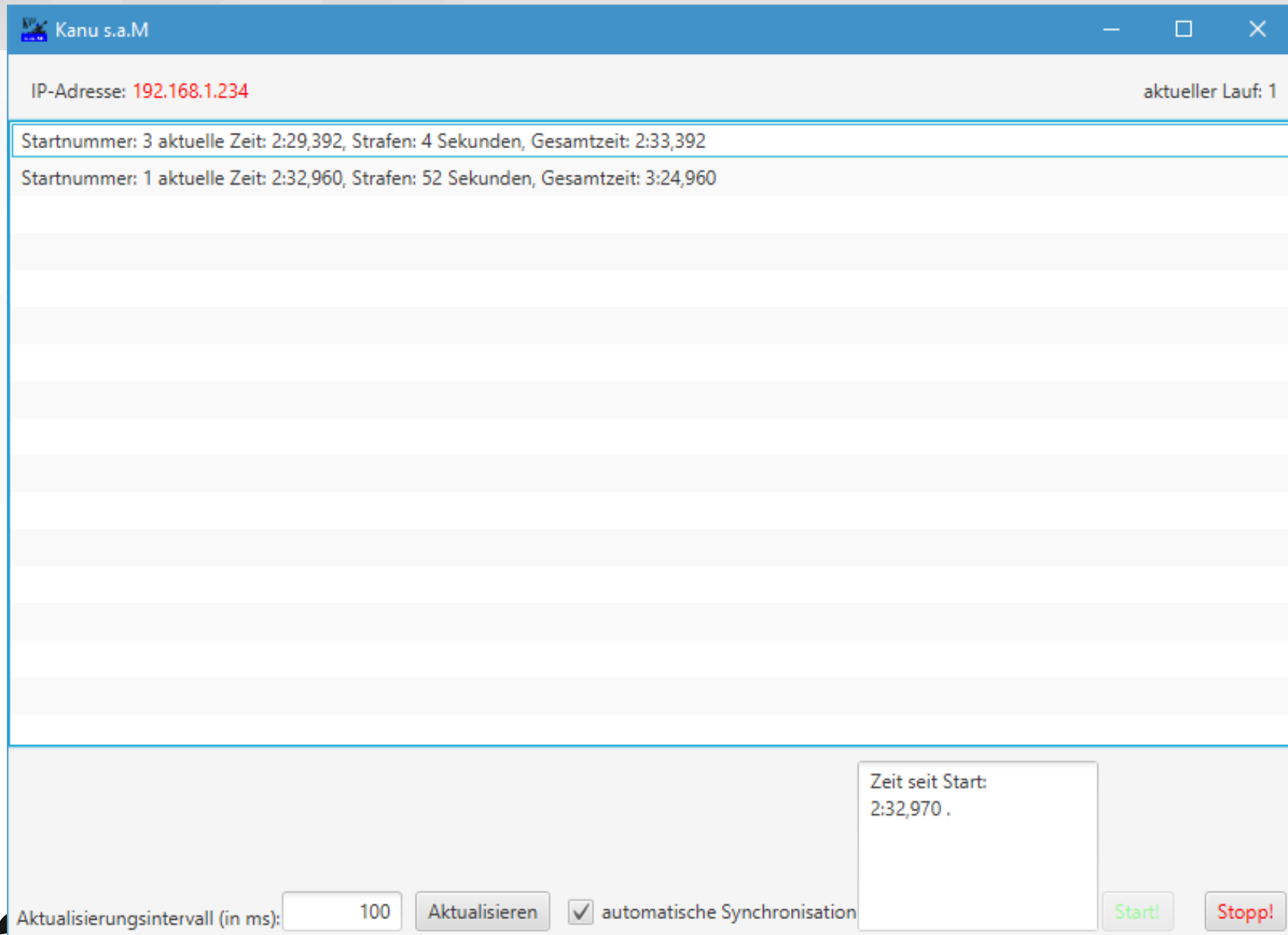
ZEIT NEHMEN!



# 3 Funktionsumfang und Einsatzszenarien



# 3 Funktionsumfang und Einsatzszenarien



The screenshot shows a software window titled "Kanu s.a.M" with a blue header bar. Below the header, the IP address "192.168.1.234" is displayed in red text on the left, and "aktueller Lauf: 1" is on the right. The main area contains a table with two rows of race data. The first row shows "Startnummer: 3", "aktuelle Zeit: 2:29,392", "Strafen: 4 Sekunden", and "Gesamtzeit: 2:33,392". The second row shows "Startnummer: 1", "aktuelle Zeit: 2:32,960", "Strafen: 52 Sekunden", and "Gesamtzeit: 3:24,960". Below the table is a large empty area with horizontal grey stripes. At the bottom, there is a control panel with a text box for "Aktualisierungsintervall (in ms)" set to "100", an "Aktualisieren" button, a checked checkbox for "automatische Synchronisation", a "Start!" button, and a "Stopp!" button. A small white box on the right side of the control panel displays "Zeit seit Start: 2:32,970".

Kanu s.a.M

IP-Adresse: 192.168.1.234 aktueller Lauf: 1

Startnummer: 3	aktuelle Zeit: 2:29,392	Strafen: 4 Sekunden	Gesamtzeit: 2:33,392
Startnummer: 1	aktuelle Zeit: 2:32,960	Strafen: 52 Sekunden	Gesamtzeit: 3:24,960

Zeit seit Start:  
2:32,970 .

Aktualisierungsintervall (in ms): 100 Aktualisieren ☒ automatische Synchronisation Start! Stopp!

Abbildung: Anzeige des Hauptprogramms während eines Laufs

# 3 Funktionsumfang und Einsatzszenarien



# 3 Funktionsumfang und Einsatzszenarien

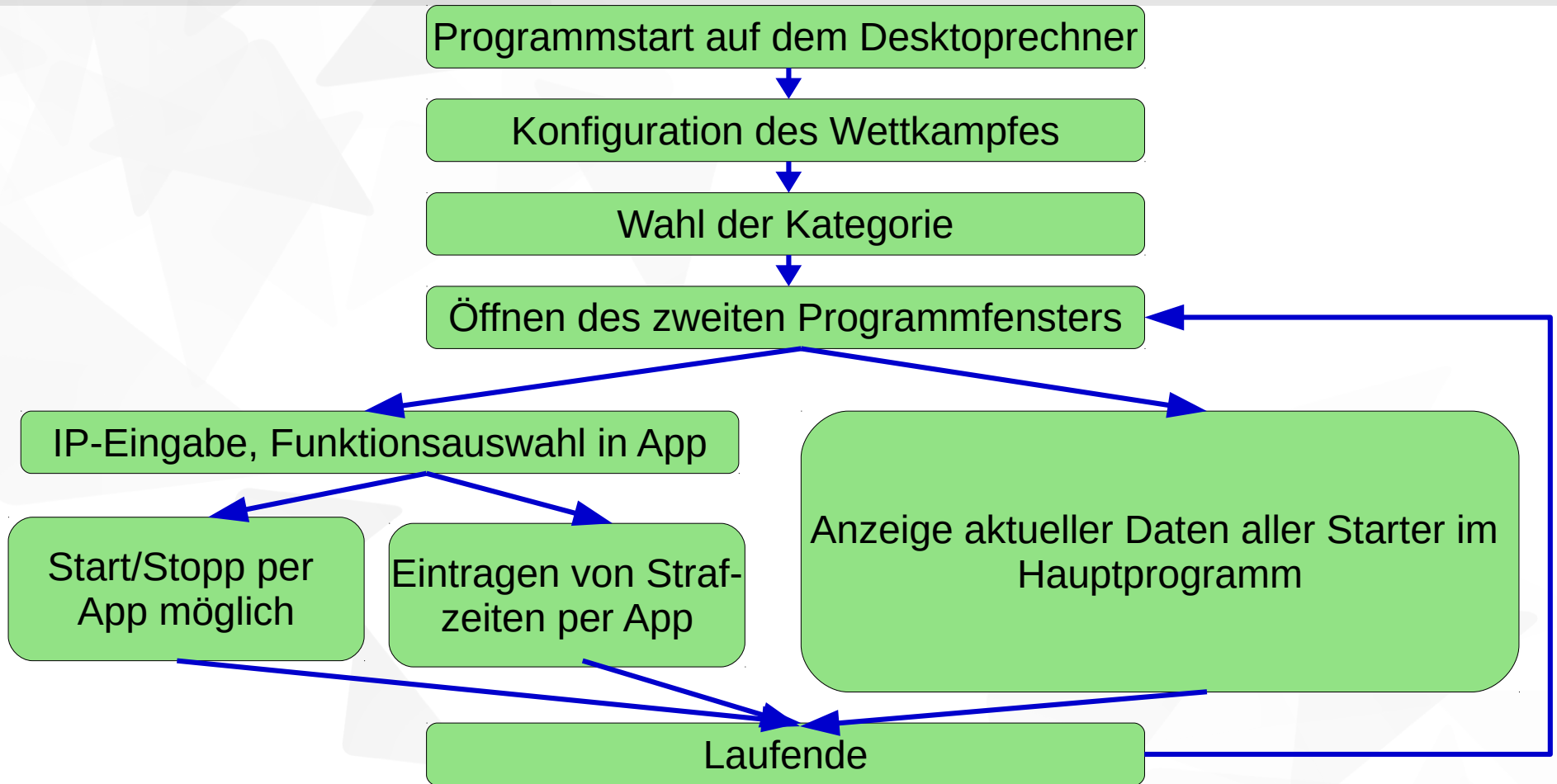
The screenshot shows the 'Kanu s.a.M' application window. At the top, the title bar reads 'Kanu s.a.M'. Below it, the IP address '192.168.1.234' is displayed in red, and 'aktueller Lauf: 1' is shown on the right. The main area contains a table with two rows of race data:

Startnummer: 3	aktuelle Zeit: 2:51,237	Strafen: 4 Sekunden	Gesamtzeit: 2:55,237
Startnummer: 1	aktuelle Zeit: 2:50,901	Strafen: 52 Sekunden	Gesamtzeit: 3:42,901

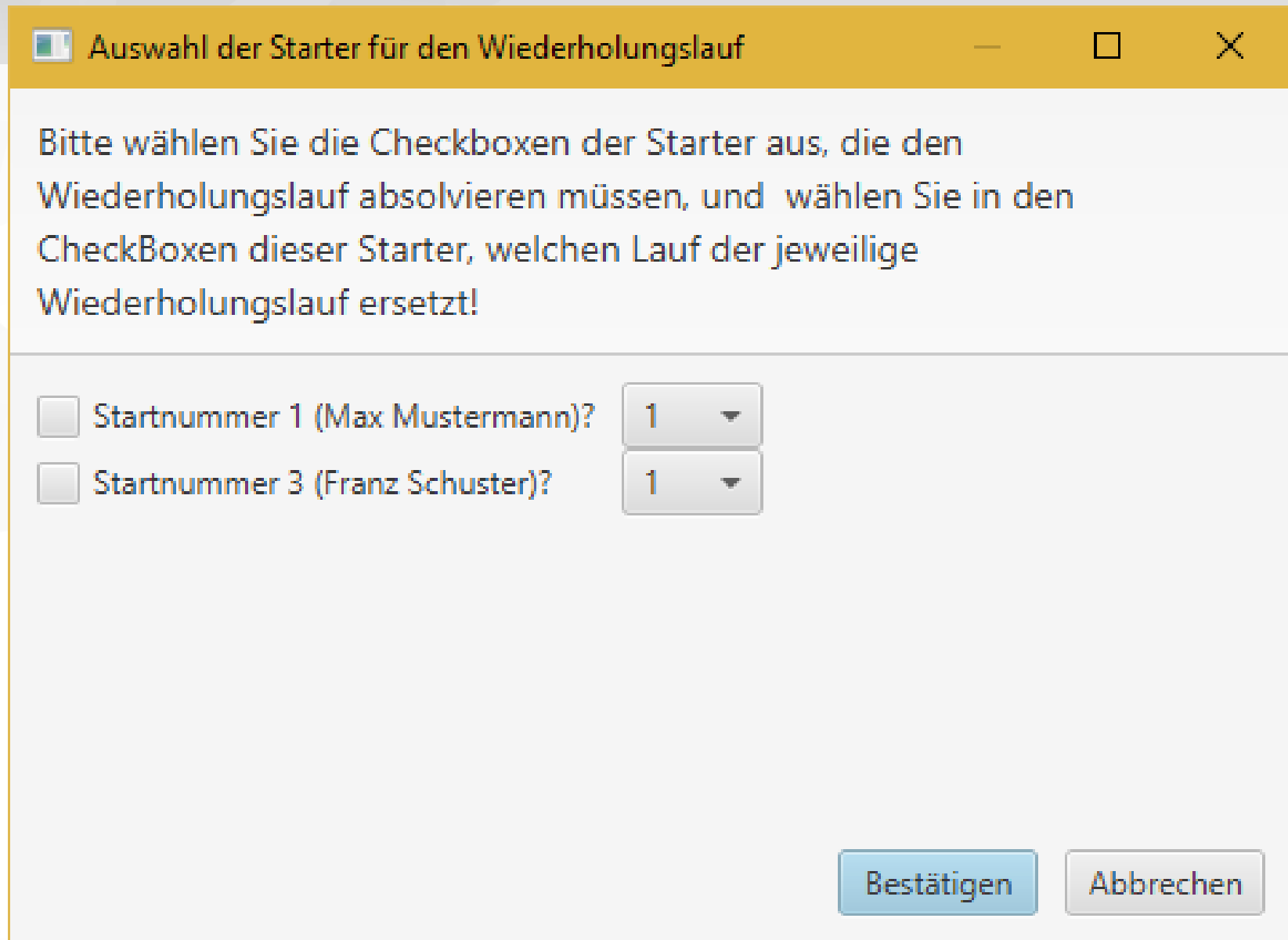
Below the table, there is a section for race controls. On the left, 'Aktualisierungsintervall (in ms):' is set to '100', followed by an 'Aktualisieren' button and a checked checkbox for 'automatische Synchronisation'. On the right, a message box says 'Der Wettkampf ist vorbei!'. Below this, there are three buttons: 'Lauf 2 starten', 'Start!' (green), and 'Stopp!' (red).

**Abbildung:** Anzeige des Hauptprogramms bei Laufende

# 3 Funktionsumfang und Einsatzszenarien



### 3 Funktionsumfang und Einsatzszenarien



Auswahl der Starter für den Wiederholungslauf

Bitte wählen Sie die Checkboxes der Starter aus, die den Wiederholungslauf absolvieren müssen, und wählen Sie in den CheckBoxen dieser Starter, welchen Lauf der jeweilige Wiederholungslauf ersetzt!

<input type="checkbox"/> Startnummer 1 (Max Mustermann)?	1 ▼
<input type="checkbox"/> Startnummer 3 (Franz Schuster)?	1 ▼

Bestätigen    Abbrechen

# 3 Funktionsumfang und Einsatzszenarien

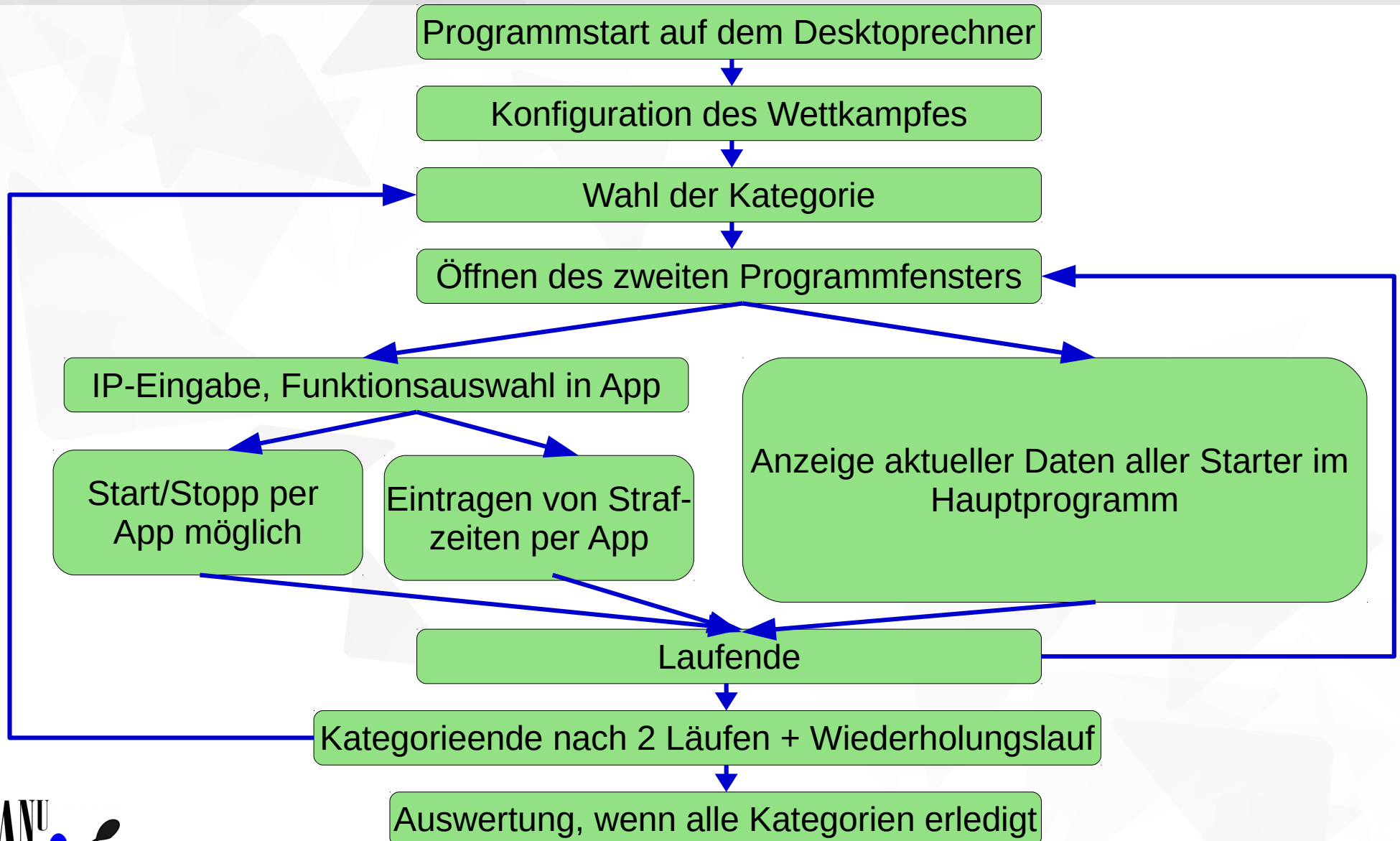


# 3 Funktionsumfang und Einsatzszenarien





# 3 Funktionsumfang und Einsatzszenarien



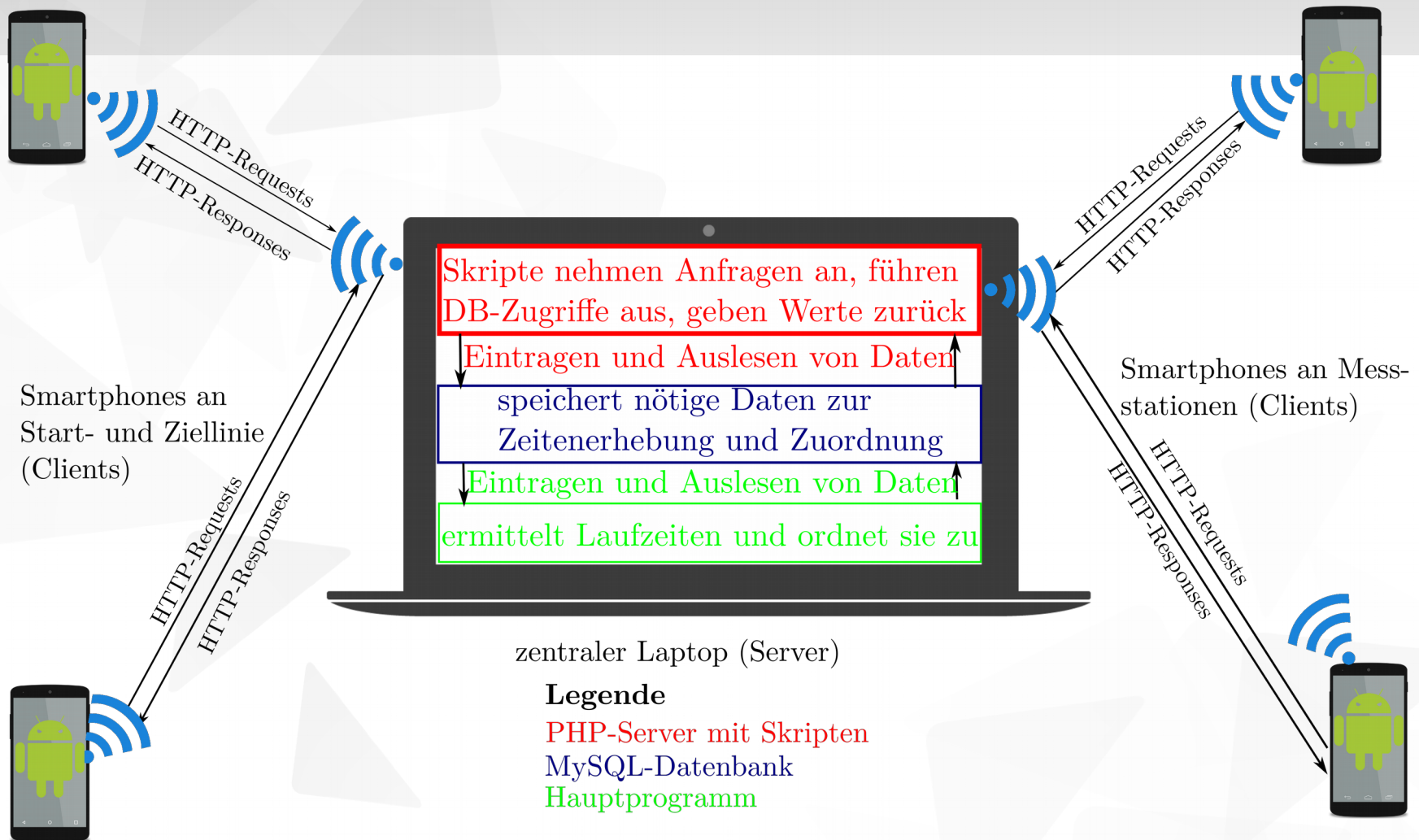
# 3 Funktionsumfang und Einsatzszenarien

	A	B	C	D	E	F		
1	Startnummer	Name	Kategorie	reine Laufzeit- Lauf 1	<u>Gesamtstrafen- Lauf 1</u>	<u>Laufzeit insgesamt- Lauf 1</u>		
2	35	<u>Herrmann Mann</u>	Pro	39,139	4	43,139		
3	15	Karl Schmidt	Pro	39,148	104	2:23,148		
4								
<div><div>&lt;</div><div>&gt;</div><div>+</div><div>Elite</div><div>Pro</div><div>Ober-Elite</div></div>								
	G	H	I	J	K	L	M	N
	Strafe Tor 1	Strafe Tor 2	Strafe Tor 3	Strafe Tor 4	Strafe Tor 5	Strafe Tor 6	reine Laufzeit- Lauf 2	<u>Gesamtstrafen- Lauf 2</u>
2	0	0	0	0	0	2	37,509	52
50	0	0	0	50	2	2	34,569	2

**Abbildung:** Anzeige der Auswertungsdatei in LibreOffice

# Entwicklungsablauf, eigene Leistungen

# 4 Entwicklungsablauf, eigene Leistungen



**Abbildung:** grundlegende Vernetzung der beteiligten Geräte im Schema

# 4 Entwicklungsablauf und eigene Leistungen

- ▼ zentral im Projekt: MySQL-Datenbank, welche während der Läufe alle Daten speichert und sowohl von den Smartphone-Apps als auch dem Hauptprogramm ausgelesen werden kann
- ▼ Datenbank wird auf Hauptrechner ausgeführt und durch MariaDB verwaltet, parallel läuft ein PHP-Server
- ▼ möglich ist dies durch Software XAMPP (auf zentralem Rechner installiert, Open-Source)

# 4 Entwicklungsablauf und eigene Leistungen



Tabelle	Aktion	Datensätze	Typ	Kollation
<input type="checkbox"/> <b>allgemein</b>	★	7	InnoDB	latin1_swedish_c
<input type="checkbox"/> <b>auswertung</b>	★	5	InnoDB	latin1_swedish_c
<input type="checkbox"/> <b>messstation_0</b>	★	0	InnoDB	latin1_swedish_c
<input type="checkbox"/> <b>messstation_1</b>	★	0	InnoDB	latin1_swedish_c
<input type="checkbox"/> <b>messstation_2</b>	★	0	InnoDB	latin1_swedish_c
<input type="checkbox"/> <b>messstation_3</b>	★	0	InnoDB	latin1_swedish_c
<input type="checkbox"/> <b>messstation_4</b>	★	0	InnoDB	latin1_swedish_c
<input type="checkbox"/> <b>messstation_5</b>	★	0	InnoDB	latin1_swedish_c
<input type="checkbox"/> <b>messstation_6</b>	★	0	InnoDB	latin1_swedish_c
<input type="checkbox"/> <b>namen</b>	★	2	InnoDB	latin1_swedish_c
<b>10 Tabellen</b>	<b>Gesamt</b>	<b>14</b>	<b>InnoDB</b>	<b>latin1_swedish_c</b>

Abbildung: Überblick über die Tabellen der Datenbank

# 4 Entwicklungsablauf und eigene Leistungen

- ▼ zentral im Projekt: MySQL-Datenbank, welche während der Läufe alle Daten speichert und sowohl von den Smartphone-Apps als auch dem Hauptprogramm ausgelesen werden kann
- ▼ Datenbank wird auf Hauptrechner ausgeführt und durch MariaDB verwaltet, parallel läuft ein PHP-Server
- ▼ möglich ist dies durch Software XAMPP (auf zentralem Rechner installiert, Open-Source)
- ▼ Datenbank speichert in den Tabellen „allgemein“ und „namen“ Daten, welche App benötigt (z.B. Zahl der Starter, Startnummern für Auswahl in den Spinnern, Zuordnung der Messtore) und die vom Hauptprogramm eingetragen wurden
- ▼ Tabellen „messstation\_0“ usw. und „namen“ speichern Strafzeiten und Start- und Zielzeiten, welche von der App eingetragen und vom Hauptprogramm synchronisiert werden
- ▼ Tabelle „auswertung“ speichert Daten, welche das Hauptprogramm zur Gesamtauswertung benötigt
- ▼ alle Tabellen außer dieser werden vor jedem Lauf bzw. jeder Kategorie zurückgesetzt, Datenbank auf Wunsch vor Programmstart oder nach Programmende geleert

# 4 Entwicklungsablauf und eigene Leistungen

- ▼ Hauptprogramm in JavaFX entwickelt in NetBeans IDE 8.2, läuft in Java Virtual Machine auf zentralem Laptop
- ▼ muss vor Start mit Starterdaten, Torkonfiguration usw. initialisiert werden, läuft danach automatisch, braucht nur einen Bediener, der jeweils nächste Kategorie/ nächsten Lauf startet
- ▼ Verbindet sich mittels Library jdbc direkt mit der Datenbank, kann über `java.sql.Connection` Lese- und Schreibabfragen ausführen



# 4 Entwicklungsablauf und eigene Leistungen

- ▼ Hauptprogramm in JavaFX entwickelt in NetBeans IDE 8.2, läuft in Java Virtual Machine auf zentralem Laptop
- ▼ muss vor Start mit Starterdaten, Torkonfiguration usw. initialisiert werden, läuft danach automatisch, braucht nur einen Bediener, der jeweils nächste Kategorie/ nächsten Lauf startet
- ▼ Verbindet sich mittels Library jdbc direkt mit der Datenbank, kann über `java.sql.Connection` Lese- und Schreibabfragen ausführen
- ▼ Leert zu Beginn die Datenbank und legt benötigte Tabellen an
- ▼ ruft während eines Laufs im eingestellten Intervall, standardmäßig jede Zehntelsekunde, Start- und Zielzeiten und Strafen aus der Datenbank ab, ordnet sie Startern zu, speichert sie lokal zwischen und gibt die aktuellen Laufzeiten der Starter aus
- ▼ Erkennt automatisch Lauf- und Kategorieende, speichert Daten in Tabelle „auswertung“, leert andere Tabellen für den nächsten Lauf
- ▼ beim Ende aller Kategorien: Programm liest gespeicherte Auswertungsdaten ein, extrahiert relevante Informationen, gibt sie mittels der Library ApachePOI in eine übersichtliche Excel-Datei aus; jede Kategorie erhält eine Mappe, dort: alle gemessenen Werte
- ▼ Link zum Sourcecode: <https://github.com/WorldofJARcraft/Kanu-s.a.M.>

# 4 Entwicklungsablauf und eigene Leistungen

- App entwickelt mit Android Studio 2.2 (offizielle IDE von Google)
- Läuft auf Smartphones in Dalvik Virtual Machine, lauffähig ab Android 2.3.3
- Android hat „von Haus aus“ keine Möglichkeit, auf MySQL-Datenbanken zuzugreifen
- aber: kann HTTP-Requests versenden, jedoch ab Android 3 nur in einem parallelen Thread (etwa AsyncTask), um die UI bei längeren Wartezeiten nicht einzufrieren
- App sendet GET-Anfragen an zentralen Laptop über IP, Apache Webserver nimmt sie entgegen

# 4 Entwicklungsablauf und eigene Leistungen

- App entwickelt mit Android Studio 2.2 (offizielle IDE von Google)
- Läuft auf Smartphones in Dalvik Virtual Machine, lauffähig ab Android 2.3.3
- Android hat „von Haus aus“ keine Möglichkeit, auf MySQL-Datenbanken zuzugreifen
- aber: kann HTTP-Requests versenden, jedoch ab Android 3 nur in einem parallelen Thread (etwa AsyncTask), um die UI bei längeren Wartezeiten nicht einzufrieren
- App sendet GET-Anfragen an zentralen Laptop über IP, Apache Webserver nimmt sie entgegen
- hinterlegte Scripte führen die Anfragen aus, entnehmen Parameter per „\$\_GET“ aus der Anfrage (etwa Startnummer und Strafzeit), führen Datenbankzugriff per „mysqli\_connect“ aus und geben HTML-Seite mit der Ausgabe oder Erfolgsmeldung aus der Datenbank zurück
- App liest Antwort per InputStream ein, sendet Anfrage bei Ausbleiben dieser erneut ab
- bei App-Start: App liest Daten wie alle Startnummern oder die Zuordnung der Messtore aus der Datenbank ein, zeigt sie auf UI an
- lässt User Parameter für das Eintragen von Zeiten (z.B. Startnummer) einstellen, sendet dann Anfrage ab
- Bei Start- und Zielzeit: PHP-Server ermittelt beim Eintreffen der Anfrage die aktuelle Zeit, trägt sie in Datenbank ein (greift auf gleiche Uhr wie Hauptprogramm zu, Smartphones ggf. nicht synchron)

- Link zum Sourcecode: <https://github.com/WorldofJARcraft/Kanu-s.a.M.-App> (PHP-Skripte im lib-Ordner des Hauptprogramms enthalten)

# Technische Spezifikation

# 5 Technische Spezifikation

## ▼ nötige Hardware:

- ▼ 1 Laptop (Windows empfohlen, Hardwareanforderungen: Windows Vista oder höher, 128 MB RAM, 126 MB freier Speicherplatz, Pentium 2 mit 266 MHz → für JRE nötig, LAN- oder WLAN-Verbindung in ein lokales Netzwerk)
- ▼ für jeden Helfer 1 Android-Endgerät mit mindestens Android 2.3.3 und WLAN
- ▼ lokales WLAN-Netzwerk möglichst im Standard WLAN-n mit ausreichender Reichweite

## ▼ nötige Software:

- ▼ XAMPP oder vergleichbar (gratis, leicht zu installieren, kompatibel ab Windows Vista)
- ▼ Java Runtime Environment (JRE); finale Version des Programms wird ohne lauffähig sein
- ▼ Office-Suite, kompatibel zum .xls-Dateiformat (etwa LibreOffice)
- ▼ 1 angelegte Datenbank, in welche das Programm schreiben kann

# Ergebnisse

# 6 Ergebnisse

- ▼ Aufbau des Zeitmesssystems ist wie gefordert einfach und kostenlos möglich
- ▼ Hardware: nur Windows-Laptop und Android-Smartphones benötigt, ggf. weitere Hardware zum Aufbauen eines ausreichend stabilen WLAN-Netzes
- ▼ Software: nur Java Runtime Environment und XAMPP benötigt, beide kostenlos
- ▼ Grundkonfiguration nötig: Aktivieren des XAMPP-Autostarts, Anlegen einer Datenbank; Rest der Vorbereitung wird vom Hauptprogramm vorgenommen
- ▼ Einrichtung des Programms ebenfalls schnell und einfach möglich, Starter werden auf Wunsch aus Excel-Datei gelesen, nur wenige manuelle Eingaben nötig
- ▼ Programm läuft nach Konfiguration fast automatisch
- ▼ Lauf- und Strafzeitermittlung und Synchronisation von Strafzeiten ist ebenfalls möglich
- ▼ Hauptprogramm erzeugt bei Programmende automatisch eine Auswertung als übersichtliche Excel-Tabelle, welche von kostenlosen Programmen wie LibreOffice geöffnet und leicht weiterverarbeitet werden kann

# 6 Ergebnisse

- Aufbau des Zeitmesssystems ist wie gefordert einfach und kostenlos möglich
- Hardware: nur Windows-Laptop und Android-Smartphones benötigt, ggf. weitere Hardware zum Aufbauen eines ausreichend stabilen WLAN-Netzes
- Software: nur Java Runtime Environment und XAMPP benötigt, beide kostenlos
- Grundkonfiguration nötig: Aktivieren des XAMPP-Autostarts, Anlegen einer Datenbank; Rest der Vorbereitung wird vom Hauptprogramm vorgenommen
- Einrichtung des Programms ebenfalls schnell und einfach möglich, Starter werden auf Wunsch aus Excel-Datei gelesen, nur wenige manuelle Eingaben nötig
- Programm läuft nach Konfiguration fast automatisch
- Lauf- und Strafzeitermittlung und Synchronisation von Strafzeiten ist ebenfalls möglich
- Hauptprogramm erzeugt bei Programmende automatisch eine Auswertung als übersichtliche Excel-Tabelle, welche von kostenlosen Programmen wie LibreOffice geöffnet und leicht weiterverarbeitet werden kann

Alles in allem wurde die Konzeption vollständig umgesetzt. Das Problem kann durch das Programm gelöst werden. Damit ist bewiesen, dass ein (minimal ungenaueres als ein professionelles) Zeitmesssystem für den Kanusport auch nahezu kostenfrei einrichtbar ist. Dieses System ist modular aufgebaut, sodass es relativ leicht um weitere Funktionen erweitert werden kann.



Originalität

## 7 Dieses Softwaresystem ist besonders, weil...

- ▼ es die Funktion von Hardware-Softwaresystemen im Wert mehrerer Kleinwagen kostenfrei und nur minimal ungenauer ausführt

## 7 Dieses Softwaresystem ist besonders, weil...

- ▼ es die Funktion von Hardware-Softwaresystemen im Wert mehrerer Kleinwagen kostenfrei und nur minimal ungenauer ausführt
- ▼ es modular aufgebaut und damit sicherer vor Abstürzen und leichter zu erweitern ist

## 7 Dieses Softwaresystem ist besonders, weil...

- ▼ es die Funktion von Hardware-Softwaresystemen im Wert mehrerer Kleinwagen kostenfrei und nur minimal ungenauer ausführt
- ▼ es modular aufgebaut und damit sicherer vor Abstürzen und leichter zu erweitern ist
- ▼ es für alle Nutzer und die weitere Entwicklung auf GitHub unter der GNU GPL 3.0 verfügbar ist

## 7 Dieses Softwaresystem ist besonders, weil...

- ▼ es die Funktion von Hardware-Softwaresystemen im Wert mehrerer Kleinwagen kostenfrei und nur minimal ungenauer ausführt
- ▼ es modular aufgebaut und damit sicherer vor Abstürzen und leichter zu erweitern ist
- ▼ es für alle Nutzer und die weitere Entwicklung auf GitHub unter der GNU GPL 3.0 verfügbar ist
- ▼ es mit Geräten aufgebaut werden kann, die in jedem Kanuverein verfügbar sind

## 7 Dieses Softwaresystem ist besonders, weil...

- ▼ es die Funktion von Hardware-Softwaresystemen im Wert mehrerer Kleinwagen kostenfrei und nur minimal ungenauer ausführt
- ▼ es modular aufgebaut und damit sicherer vor Abstürzen und leichter zu erweitern ist
- ▼ es für alle Nutzer und die weitere Entwicklung auf GitHub unter der GNU GPL 3.0 verfügbar ist
- ▼ es mit Geräten aufgebaut werden kann, die in jedem Kanuverein verfügbar sind
  - ▼ es in der Anwendung variabel ist und in vielen verschiedenen Wettkämpfen und Vereinen eingesetzt werden kann

# KANU

Kanu s.a.M.



Ein Beitrag zum **Inverso** Softwarewettbewerb in  
der Kategorie **Schüler**

# s.a.M.