

# Step-Projekt des Teams 01

## CGI-Webanwendung zur Analyse und Darstellung von Rhodes-Daten

Natalia Karachevtseva , Ibrahim Khalil Sakka , Mustafa Kemal Kivrakoglu , Bjarne Lührs , Muhammed Enes Bal , Christoph Gebauer

### Einführung in unser Projekt

Unser Team hat die Aufgabe, eine dynamische Webanwendung zur Visualisierung von AIS-Daten (rhodes – 194.94.217.72) zu entwickeln. Das Automatische Identifikationssystem (AIS) wird in der Schifffahrt eingesetzt und ermöglicht die drahtlose Kommunikation zwischen Schiffen und Bodenstationen. Dabei werden Informationen wie Identifikation, Position, Kurs und Geschwindigkeit ausgetauscht.

AIS-Daten liegen in Form von AIS-Nachrichten vor, die spezifische Informationen über ein Schiff enthalten. Ziel unseres Projekts ist es, eine Plattform bereitzustellen, auf der Nutzer die Schiffsdaten (Name, MMSI, IMO, Zielort, Position) der letzten sieben Tage für ein bestimmtes Gebiet abrufen können.

### Zielstellung

Das Projektziel ist die Entwicklung einer CGI-Webanwendung zur Anzeige von Echtzeitdaten, die kontinuierlich von unserem AIS-Empfänger auf Rhodes empfangen werden. Die Anwendung soll ein benutzerfreundliches Interface bieten, das die Daten jedes Schiffes auf separaten Seiten darstellt. Dabei dürfen nur die während der Veranstaltung vorgestellten Technologien verwendet werden. Alle Teammitglieder müssen in der Lage sein, alle Aspekte der Umsetzung des Projekts detailliert zu erklären, und auf Anfrage müssen zentrale Skripte während der Präsentation oder in späteren Diskussionen zugänglich und nachvollziehbar sein. Zusätzlich wird ein Beobachtungsmodul entwickelt, das den AIS-Datenstrom kontinuierlich überwacht. Sollte innerhalb der letzten Minute keine neuen Daten eingegangen sein, wird das Empfangsprogramm automatisch neu gestartet, um die Datenversorgung sicherzustellen.

### Projektidee, Umsetzung, Arbeitsweise und Darstellung

**1. Idee :** Um eine benutzerfreundliche Lösung zur Anzeige relevanter Schiffsinformationen zu entwickeln, haben wir zunächst analysiert, welche Daten für unsere Nutzer am wichtigsten sind. Daraus entstand die Idee einer Webseite, die die wichtigsten Informationen zu Schiffen übersichtlich in Tabellen darstellt. Die Daten werden für die letzten sieben Tage bereitgestellt, sodass stets aktuelle Informationen verfügbar sind.

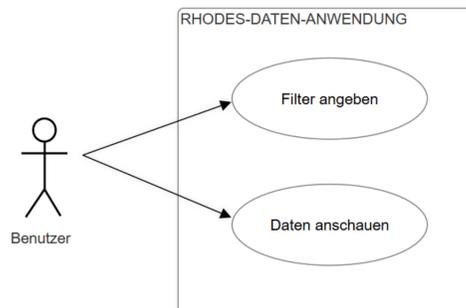


Abbildung 1: Use Case Diagramm

**2. Arbeitsweise :** Auch wenn wir uns selber noch nicht daran gehalten haben, haben wir dennoch eine ideale Arbeitsweise, um möglichst effizient zu arbeiten. In unserem Team-Ordner in Hopper befindet sich der Ordner „entwürfe“, dort haben wir verschiedene Versionen unserer Website inklusive eines Skriptes namens „commit.sh“. Unsere Arbeitsweise sieht vor, dass wir in diesem Ordner an Skripten arbeiten und sie dann mithilfe der commit.sh auf den docker schicken, wo das Skript dann ersetzt wird. Wichtig ist vorher, dass man mit scp den aktuellen Status der Anwendung in den entwürfe step Ordner lädt, um die alte Version zu speichern. Das gibt einem dann die Möglichkeit auf einen älteren Zustand zurückzugehen.

```
1 datei="$1"
2 if test -e $1; then
3   tar -cf - $datei | sudo -i -u
4     ↪ step-2024-team01 bash -c "ssh mydocker
5     ↪ 'tar -xf - -C /usr/lib/cgi-bin/"
6   echo "Übertragung abgeschlossen"
7 else
8   echo "Fehler: Datei nicht gefunden!"
9 fi
```

Listing 1: commit.sh

**3. Umsetzung :** Unsere Lösung basiert auf drei Hauptskripten, die nahtlos zusammenarbeiten:

- \* **Watcher :** Dieses Skript ruft täglich aktuelle Schiffsdatensätze vom Rhodes-Server ab und speichert sie in einem strukturierten Dateiverzeichnis. Jede Datei trägt das jeweilige Datum als Namen, um eine geordnete Archivierung zu gewährleisten.

- \* **Hauptsript :** Die gesammelten Daten werden verarbeitet und in einer Tabellenstruktur dargestellt. Nutzer können gezielt nach Datum, Typennummer oder MMSI-Nummer filtern. Zudem gibt es eine Exportfunktion für den Download als CSV-Datei.

- \* **Löscher :** Dieses Skript entfernt ältere Datensätze automatisch. So stehen stets nur die letzten sieben Tage zur Verfügung, um die Aktualität der Daten zu sichern. Ein Sicherheitsmechanismus stellt sicher, dass nur veraltete Dateien gelöscht werden.

```
1 if test $zeilen_alt == $zeilen ; then
2   echo "$(date +%H%M%S)" >> "$fehler"
3   kill $(cat "$letztes_pid")
4   ncat -e ./eins.sh rhodes 8082 &
5   echo "$!" > "$letztes_pid"
6 else
7   echo "$(date +%H%M%S)" >> "$erfolg"
8 fi
```

Listing 2: watcher.sh

**4. Darstellung :** Die Hauptseite der Webseite bietet verschiedene Filteroptionen sowie eine täglich aktualisierte Grafik, die die Anzahl der empfangenen Nachrichten pro Tag visualisiert. So können Nutzer auf einen Blick erkennen, wie viele Schiffsdatensätze in den letzten sieben Tagen erfasst wurden.

Zusätzlich ermöglicht die Filterfunktion eine gezielte Suche nach Datum, MMSI-Nummer oder Typennummer, um schnell relevante Daten zu finden.

Zudem haben wir einen einfachen Lasttest mit *curl* durchgeführt.

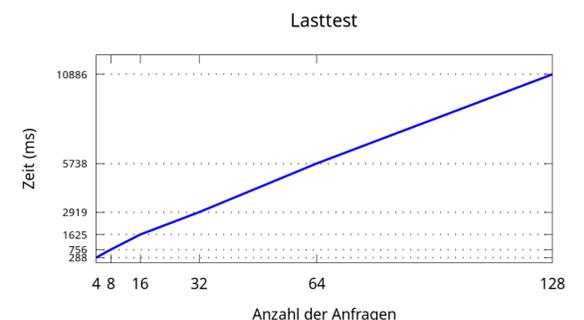


Abbildung 2: Lasttest

Unsere Webanwendung bietet somit eine effiziente, benutzerfreundliche Möglichkeit zur Analyse und Darstellung aktueller Schiffsbewegungen.

### Schluss

Mit unserer Webanwendung haben wir eine benutzerfreundliche Lösung zur Analyse und Visualisierung von AIS-Daten geschaffen. Sie ermöglicht es den Nutzern, Schiffsbewegungen in Echtzeit zu verfolgen und gezielt nach wichtigen Informationen zu suchen. Dank der Automatisierung der Datenerfassung und Verarbeitung bleibt die Anwendung immer aktuell.

Die Umsetzung unseres Projekts hat uns nicht nur technische Fähigkeiten im Bereich der Webentwicklung vermittelt, sondern auch unsere Teamarbeit gestärkt. Herausforderungen wie die Datenverarbeitung und die Gestaltung eines intuitiven Interfaces haben wir gemeinsam erfolgreich gelöst.

Unsere Anwendung ist uns gelungen und bietet eine solide Grundlage für zukünftige Erweiterungen.