Team-09-Poster

Hochschule Bremerhaven

Unser AIS-Daten Verarbeiter

Oliver Wendt; Emircan Aslan; Kevin Ziegler; Florian Radtke; Dominic Wittop

Einleitung

Im Rahmen des Studiengangs Wirtschaftsinformatik und Informatik legten wir im Modul STEP nicht nur den Grundstein für unser nachfolgendes Studium, sondern lernten auch durch einen praxisnahen Bezug das Zusammenarbeiten als Team. Wir erlangten essenzielle Fähigkeiten und Kenntnisse, die als Basis für die erfolgreiche Umsetzung unserer ersten Projektarbeit dienten. Zu den erlernten Fähigkeiten zählen unter anderem der Umgang mit einem Terminal und mit Unix. Mit wöchentlichen Aufgaben und Berichten haben wir die Möglichkeit bekommen, die in den Veranstaltungen gelernten Fähigkeiten zu dokumentieren, anzuwenden und einzuprägen.

Das Projekt

Die Benutzer*innen können beim Aufrufen unserer dynamischen Webanwendung, AIS-Daten abrufen und damit Interagieren. Hierbei kann er/sie die Daten nach verschiedenen Typen oder nach ihrem Status filtern. Zusätzlich besteht die Option, die Anzahl der abgerufenen Datensätze im Bereich von 100 bis 1000 zu variieren. Nach dem Absenden des Formulars werden die Parameter über den Query-String an das entsprechende Skript übermittelt.

Für den Sendetypen-Filter wird das Skript “type-filter.sh”verwendet. Dieses greift auf die neuesten Daten aus der zugehörigen Datei zu, die bereits zuvor von unserem Skript “aisdaten-spliter.sh“ aus der “aislog-complete.log”gelesen und in die entsprechenden “aistype-.log”Dateien sortiert wurden.

Die erhaltenen Daten werden auf einer neuen Webseite in einer Tabelle mit wichtigen Eigenschaften angezeigt. Die MMSI-Nummern sind bei den Typen 1, 3 und 5 als Links in der Tabelle generiert. Beim Anklicken eines Links wird das Skript “detail-filter.sh“ ausgeführt, das eine neue Webseite generiert und weitere Details zum Schiff mit der entsprechenden MMSI-Nummer anzeigt.

Beim Absenden des Status-Filters wird das Skript “status-filter.sh“ ausgeführt, das in den Dateien “aistype-1.log“ und “aistype-3.log“ nach den übermittelten Statuswerten filtert. Der Filter ist nur auf diese beiden Typen begrenzt, da nur sie den Status übermitteln.

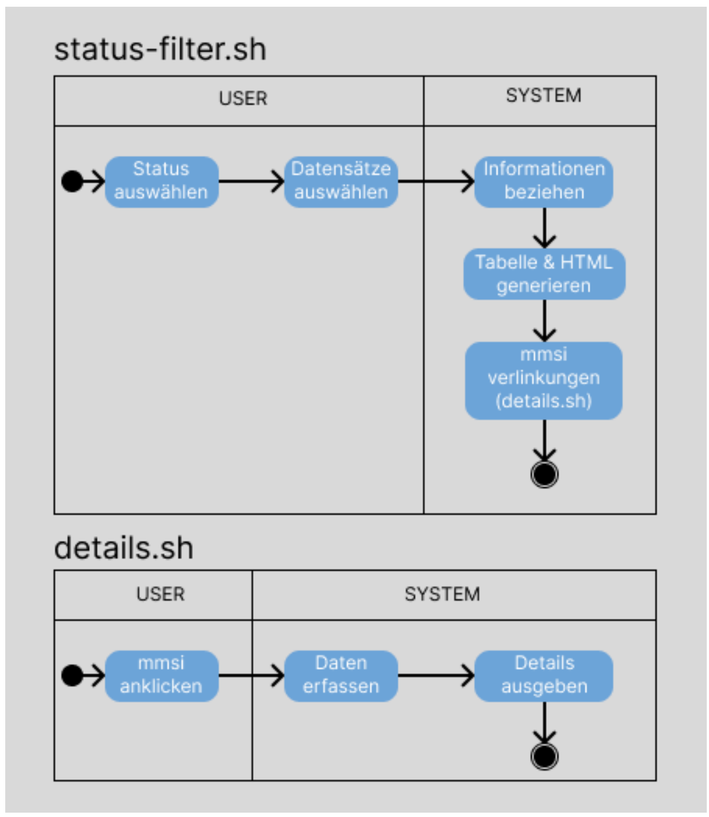

Abbildung 1: UML Aktivitätsdiagramm

Umsetzung

```
#Aistype-1.log
grep -m 1 $mmsi $file/aistype-1.log |while IFS='|' read -r date type mmsi status stats_text turn speed accuracy lon lat course heading second maneuver maneuver_text raim radio; do
echo "Daten abgerufen am: $date
        <br>
        MMSI-Nummer: $mmsi
        <br>
        Status: $status
        <br>
        Längengrad: $lon
        <br>
        Breitengrad: $lat
        <br>
        Funkfrequenz: $radio
        <br>"
done
#Aistype-5.log
grep -m 1 $mmsi $file/aistype-5.log |while IFS='|' read -r date type mmsi imo callsign shipname shiptype shiptype_text to_bow to_stern to_port to_starboard epfd eta epfd_text month day hour minute draught destination dte; do
echo "Schiffname: $shipname
        <br>
        Schifftyp: $shiptype_text
        <br>
        Ziel: $destination
        <br>
        Ankunft am Ziel: $day.$month"
done
```

Listing 1: details.sh

Last-Test

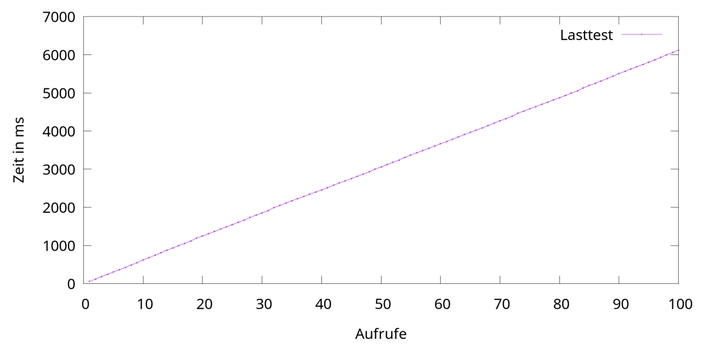

Abbildung 2: Unser Last-Test

Fazit unseres Projektes

Unser Projekt hat sich als äußerst energieeffizient erwiesen und ist in der Lage, eine ausgewogene Auslastung auch bei einer Vielzahl von Anfragen aufrechtzuerhalten. Selbst bei 100 Anfragen beträgt die Verzögerung unseres Projekts weniger als 7000ms, was auf eine stabile Programmstruktur mit geringer Belastung hinweist. Wie es im Artikel betont wurde “ Eine umweltfreundlichere Programmierung der Software wird auch als Green Coding bezeichnet. Damit gemeint ist etwa, dass Websites so programmiert werden, dass Bilder automatisch skaliert werden oder dass Endgeräte nur in sinnvollen Zeitabständen eine Anfrage an Datenbanken nach einem Update schicken.”[1] Unsere Priorität lag auf der Energieeffizienz, um ein vorbildliches Beispiel für Green Coding zu setzen. Durch die Vermeidung komplexer und zeitaufwändiger Skripte konnten wir ein positives Ergebnis bei den Lasttests erzielen.

Schluss

Step hat uns in diesem Semester einen ersten Eindruck von der Arbeit mit Computern gegeben. Wir haben in dem letzten halben Jahr viele Möglichkeiten kennengelernt, die uns helfen in der Unix Umgebung zurechtzukommen und dort eigene Programme zu schreiben. Des Weiteren haben wir die Erfahrung gemacht, in Teams zu arbeiten, was so wohl Schwierigkeiten, aber auch viele Vorteile brachte. Zusammenfassend kann man sagen, dass Step, besonders durch das Engagement der Tutoren, zu einem guten Einstieg ins Studium geführt hat.

1 Referenzen

[1] S. Winkler, J. Günther und R. Pfennig, *Nachhaltige Digitalisierung oder Nachhaltigkeit durch Digitalisierung?* Springer, 2023, S. 1–22.