

Einstieg und Planung

Im Rahmen des Moduls 'Einführung in die Informatik' haben wir uns intensiv mit der Entwicklung dynamischer Webanwendungen auseinandergesetzt. Die erworbenen Kenntnisse vertieften wir durch eine praxisorientierte Projektarbeit. Unser Ziel in diesem Projekt, dem Schiffsradar, bestand darin, die Schiffsdaten in einer CSV-Datei zu speichern und eine benutzerfreundliche HTML-Seite zu erstellen. Auf dieser Seite kann der Benutzer mithilfe einer Tabelle bequem die Schiffsdaten der letzten drei Tage abrufen. Darüber hinaus ermöglicht es die Anwendung, die Uhrzeit auszuwählen, um die Aktivitäten der Schiffe zu einem bestimmten Zeitpunkt anzuzeigen. Eine weitere Funktion besteht darin, durch Klicken auf die MMSI (Schiffsnummer) detaillierte Signale dieses spezifischen Schiffs tabellarisch darzustellen. Um sicherzustellen, dass die Schiffsdaten auf dem neuesten Stand sind, erfolgt eine stündliche Aktualisierung, bei der die aktuellen Informationen in neuen CSV-Dateien abgespeichert wurden. Automatisch gelöscht werden dabei ältere CSV-Daten, die älter als drei Tage sind.

Modellierung in der UML

Das Aktivitätsdiagramm verschafft einen guten Überblick über den genauen Ablauf des Portals. Es wird von zwei Abteilungen - Benutzer und System - bearbeitet. Nach dem Aufruf der Seite durch den Benutzer wird die Seite vom System angezeigt. Anschließend hat der Benutzer die Möglichkeit, das Datum und die Uhrzeit auszuwählen, zu denen er überprüfen möchte, welche Schiffe in dieser Zeit aktiv waren. Dann ruft das System die Daten ab und erzeugt eine Tabelle. Die Tabelle wird dann mit allen Informationen angezeigt, welche der Benutzer ausgewählt hat. Es besteht die Möglichkeit, eine weitere Liste anzeigen zu lassen, indem der Benutzer auf die MMSI (Schiffsnummer) klickt. Dann werden alle Signale zu diesem Schiff, die von ihm geschickt wurden, in der Tabelle angezeigt.

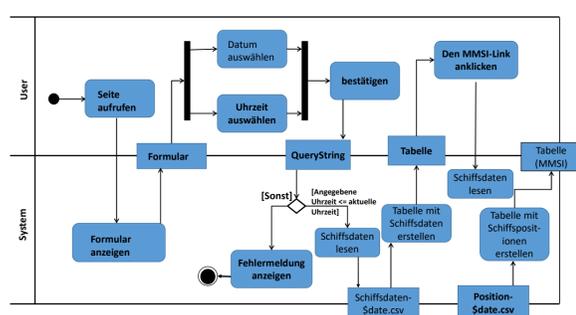


Abbildung 1: Aktivitätsdiagramm

Umsetzung

Die Implementierung von Schiffsradar Bremerhaven wurde größtenteils mit Bash umgesetzt, womit auch die CSV-Dateien erstellt wurden. Daneben wurden auch weitere Technologien wie GnuPlot zur Erstellung des Diagramms, HTML zur Erstellung der Webseite, CSS zur Bearbeitung des Designs der Webseite und LaTeX zur Generierung des PDF-Dokuments verwendet. Wir haben uns entschieden, ein Projekt mit den Werkzeugen zu entwickeln, die wir gelernt haben. Da es sich um AIS-Daten arbeiten, haben wir uns dafür entschieden, Typ 3 und Typ 5 zu extrahieren und in zwei verschiedenen CSV-Dateien zu speichern.

Von Typ 3 extrahieren wir Datum, Typ, MMSI, IMO, Geschwindigkeit, Längengrad und Breitengrad, und diese werden in 'position.csv' gespeichert. Bei Typ 5 handelt es sich um die restlichen Informationen wie Datum, Typ, MMSI, IMO, Schiffsnamen, Schiffstyp und Ziel, die dann in 'table.csv' gespeichert werden.

Die Schiffsdaten werden nur für die letzten drei Tage (aktueller Tag und zwei Tage zuvor) angezeigt. Auf der Startseite hat der Benutzer die Möglichkeit auszuwählen, von welchem Tag und von welcher Uhrzeit er die Daten aufrufen möchte. Wenn er eine Uhrzeit auswählt, welche über die aktuelle hinausgeht, wird er folgende Meldung vom System erhalten: 'Unser System befindet sich derzeit in der Phase der Datenverarbeitung'. Die CSV-Dateien werden in einem Verzeichnis gespeichert und beim Start des Programms werden sie aufgerufen und durchgeführt.

```
1 tac ~/datei/unsorted/$positiondatei | sort
  → -t'|' -k2,2 -u >>
  → ~/datei/bin/position.csv
2 tac ~/datei/unsorted/$tabledatei | sort
  → -t'|' -k2,2 -u >> ~/datei/bin/table.csv
3 cat ~/datei/bin/position.csv | while
  → IFS='|' read datum mmsi rest ;do
4 cat ~/datei/bin/table.csv | while IFS='|'
  → read DATUM MMSI REST;do
5 if [ "$MMSI" = "$mmsi" ];then
6 echo "$datum|$mmsi|$rest|$REST"
  → >>~/datei/sorted/schiffsdaten-$einst
  → undevor.csv
7 fi
8 done
9 done
```

Listing 1: sortierung.sh

Des Weiteren haben wir einen Lasttest durchgeführt, um die Ausführungsdauer zweier unterschiedlicher Implementierungen zu vergleichen: eine basierend auf einer While-Schleife und die andere unter Verwendung von regulären Ausdrücken (Regex). Wir wollten verstehen und veranschaulichen, wie sich die Dauer der Programme in Abhängigkeit von der Anzahl der Benutzer entwickelt. Die analysierten Ergebnisse lieferten unerwartete Erkenntnisse im Vergleich der Ausführungsdauern zwischen der

While-Schleifen- und der RegEx-basierten Implementierung. Die gemessenen Unterschiede in der Dauer könnten auf externe Faktoren zurückzuführen sein, die möglicherweise nicht vollständig berücksichtigt wurden. „In der heutigen Zeit ist nachhaltige Programmierung von zunehmender Bedeutung, um Ressourcen effizient zu nutzen und langfristig zuverlässige Lösungen zu gewährleisten.“ [1] Unser Ziel der Lasttests war es, die bessere Option zwischen zwei Ansätzen zu finden. Nach unserer Analyse stellten wir jedoch fest, dass es keinen deutlichen Unterschied gibt. Dies könnte eine Folge dessen sein, dass wir mit einer begrenzten Datenmenge arbeiten.

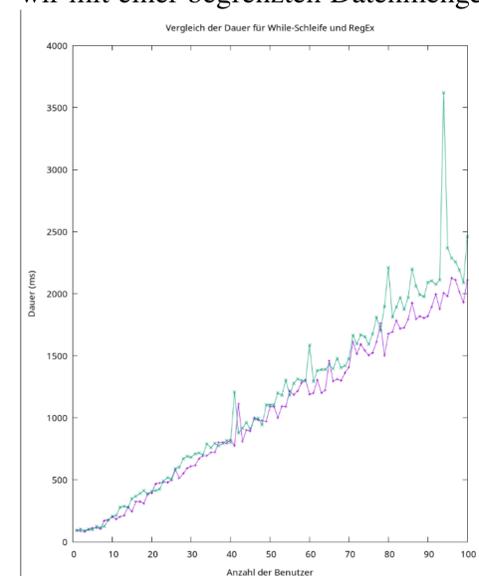


Abbildung 2: Gnuplot von Lasttest

Schluss

Die Entwicklung vom Schiffsradar Bremerhaven lief am Anfang nicht so gut, da nicht alle Teammitglieder auf dem gleichen Stand waren. Das Projekt konnten wir leider nicht komplett umsetzen, wie es geplant wurde, da wir nicht ausreichende Werkzeuge besaßen. Ein Teil unseres Planes war es, ein Gnuplot mit einer Karte von Bremerhaven als Hintergrund zu haben und die Schiffe sollten als Punkte auf der Karte gezeigt werden. Die Punkte haben wir hinbekommen, allerdings war es zu schwer, die Karte als Hintergrund zu machen. Die STEP Lehrveranstaltung war ein interessanter und abwechslungsreicher Einstieg in unser Studium. Das Projekt hat uns geholfen, unser Wissen zu verbessern und wie wir als Team alles Planen und besser organisieren können. Diese Erfahrungen haben nicht nur unser Teamzusammenhalt gestärkt, sondern uns auch einen ersten Einblick in die Arbeit in einem kooperativen Umfeld gewährt.

Referenzen

- [1] S. Winkler, J. Günther und R. Pfennig, „Nachhaltige Digitalisierung oder Nachhaltigkeit durch Digitalisierung?“ *HMD Praxis der Wirtschaftsinformatik*, S. 1–22, 2023.