



# Entwicklung Online Fahrgastinformationssystem für das EBL

Im Rahmen eines Softwarepraktikums wird einer Gruppe von Studenten der Informatik die Aufgabe gestellt, ein Online Fahrgastinformationssystem zu erstellen, das innerhalb des Eisenbahnbetriebslabors von überall aus aufgerufen werden kann.

## Einleitung

Im Eisenbahnbetriebslabor findet im Rahmen von Forschung und Lehre eine gegenständliche Simulation des Eisenbahnbetriebs mit Personen- und Güterverkehr statt. Die Praktikumssteilnehmer lernen dabei wie der Eisenbahnbetrieb funktioniert und welche Tätigkeiten dafür notwendig sind. Um das Lernen realistischer zu gestalten, soll für das Labor als Weiterentwicklung ein Fahrgastinformationssystem aufgebaut werden.

## Aufgabenstellung

Im Rahmen des Softwarepraktikums soll eine Webbasiertes Fahrgastinformationssystem (FIS) für das Eisenbahnbetriebslabor entwickelt werden. Über die Webseite sollen die Benutzer:

- eine Abfahrts- und Ankunftsanzeige zu einem bestimmten Bahnhof aufrufen können
- einen Zuglauf für einen ausgewählten Zug anzeigen können.

In beiden Anzeigen sollen Echtzeitinformationen, bestehend aus Gleisänderungen, Verspätungsminuten und Textmeldungen, mit angezeigt werden.

Das System sollte im abgeschlossenen Labornetzwerk auf einen internen Apache2-Webserver laufen. Seine Fahrplan- und Echtzeitdaten soll das Fahrgastinformationssystem von einem Laborinternen Fahrplanserver bekommen und auch aus einer XML-Datei auslesen können.

Als Orientierungshilfe kann von der Online-Abfahrtsanzeige der Deutschen Bahn ausgegangen werden:

<http://reiseauskunft.bahn.de/bin/bhftafel.exe/> [http://reiseauskunft.bahn.de/bin/bhftafel.exe/]

## Hauptsächliche Nutzungsszenarien

Durch die Internetseite sollen die Benutzer Informationen zu Bahnhöfen und Zügen abfragen und können und sich diese in tabellarischer Form anzeigen lassen. Die Interaktion mit dem Benutzer beschränkt sich daher auf den Aufruf der Seite und die Eingabe/Auswahl der Züge/Bahnhöfe, der Uhrzeit und der Zugtypen. Die Nutzungsszenarien lassen sich daher auf folgende 3 Szenarien, sowie ein Administratives Szenarium eingrenzen.

### Aufruf der Internetseite

Durch Eingabe der Internetadresse

`http://<IP-des-Webservers>/fis`

bekommt der Benutzer eine Auswahlseite angezeigt, in der er auswählen kann, ob er:

- eine Abfahrtstafel,
- eine Ankunftstafel, oder
- einen Zuglauf angezeigt bekommt.

Je nach Auswahl muss der Benutzer dann weitere Daten eingeben:

- Bei der Abfahrts- und bei der Ankunftstafel muss der Nutzer den Bahnhof, die Anfangszeit (ab dieser werden Züge angezeigt) sowie die Endzeit oder Zeitdauer (bis zu der die Züge angezeigt werden) eingeben.
- Beim Zuglauf muss der Benutzer eine Zugnummer eingeben oder aus einer Combobox auswählen.

Nach Auswahl und Eingabe drückt der Benutzer auf einen Button „Anzeigen“ (Info zur Umsetzung: ein Button zur Anzeige der Auswahl oder auch einzelne „Anzeigen“-Buttons bei den Auswahlen, oder auch ein selbst vorgeschlagenes Button-loses Verfahren ist möglich) und es wird im Browser eine neue Seite für die Abfahrten, Ankünfte oder Zuglauf geladen und angezeigt.

## **Wechsel von Abfahrt-/Ankunftsdarstellung zu einem Zuglauf**

Wenn der Benutzer auf der Seite mit den Abfahrten-/Ankünften ist, kann er auf eine Zugnummer klicken. Nach dem Klick wird die aktuelle Seite durch eine neue Seite mit dem Zuglauf für die angeklickte Zugnummer geladen und angezeigt.

## **Wechsel von einer Zuglaufdarstellung zur Abfahrtsdarstellung**

Wenn dem Benutzer eine Seite mit den einem Zuglauf angezeigt wird, kann er auf einen Bahnhof innerhalb des Zuglaufes klicken. Nach dem Klick wird die aktuelle Seite durch eine Abfahrtsdarstellung für den angeklickten Bahnhof ersetzt.

## **Ändern der Konfiguration**

Die Konfiguration soll als INI-Datei in einem Unter- oder Überverzeichnis liegen. Wenn der Administrator die Konfiguration des FIS ändern will, ruft er die INI-Datei in einem beliebigen Texteditor auf, ändert die Werte und speichert diese ab. Danach startet der das FIS neu.

---

## **Aufbau der Darstellung**

Die hier benannten Beschreibungen sollen bei der Erstellung der Seiten berücksichtigt werden. Alle nicht genannten Punkte können, soweit nicht im weiteren Verlauf des Dokumentes genannt, nach eigenem Ermessen erstellt und dargestellt werden.

### **Kopf- und Fußzeile**

Der allgemeine Aufbau der Internetseiten soll so sein, dass alle Seiten immer eine gleiche Kopf und Fußzeile haben. In der Kopfzeile wird links ein änderbares Logo angezeigt, in der Mitte ein seitenabhängiger Text, am rechten Rand die aktuelle Laboruhrzeit. Die Fußzeile soll rechtsbündig einen, in der Konfiguration fest einstellbaren Text, sowie linksbündig die Programmversion und den Verbindungsstatus zum Fahrplanserver anzeigen.

### **Auswahlseite**

Die Auswahlseite soll dem Benutzer die Möglichkeit geben, zu einer der 3 Unterseiten zu gehen. Wie die Auswahlmöglichkeit und die dazugehörigen auswahlspezifischen Einstellungsmöglichkeiten aussehen und angeordnet

sind, ist den Entwicklern frei überlassen. In der Kopfzeile soll im seitenabhängigen Text „Fahrgastinformationssystem“ angezeigt werden.

## Abfahrtsanzeige

Diese Seite soll dem Benutzer eine Tabelle mit folgenden Spalten in genannter Reihenfolge anzeigen:

1. Zeit
2. Zug
3. Richtung / Unterwegshaltestellen
4. Gleis
5. Aktuelles

In der Tabelle soll ein Zug pro Zeile angezeigt werden, mit den folgenden Informationen in der jeweiligen Spalte:

1. Planmäßige Abfahrtszeit (SollZeit) des Zuges an diesem Bahnhof
2. Gattung (ICE, RE, RB, ...) und Zugnummer
3. In dieser Spalte wird eine Auflistung aller nachfolgenden Bahnhöfe inkl. der planmäßigen Abfahrtszeit angezeigt, in Form von „<Bahnhofsname> <Abfahrtszeit> - “. Der „-“ soll den vorherigen Bahnhof vom nächsten trennen. Der Endbahnhof soll in einer abweichenden Schriftart oder Schriftfarbe hervorgehoben werden.
4. Das planmäßige Abfahrtsgleis in diesem Bahnhof
5. Die Aktuelle Verspätung in Minuten aus IstZeit-SollZeit, das geänderte Abfahrtsgleis und ein übermittelter Text.

Im Bereich der Seite soll auch eine Filtermöglichkeit angezeigt werden, in der die angezeigten Züge nach Ziel, Zuggattung und Zeitraum gefiltert werden können. Nach der Filterung sollen die Änderungen direkt in der Tabelle angewendet werden. Die Position der Filterelemente kann von den Entwicklern frei gewählt werden.

In der Kopfzeile soll der seitenabhängige Text den Namen des Bahnhofs anzeigen.

## Ankunftsanzeige

Die Ankunftsanzeige soll die gleiche Tabelle anzeigen, wie die Abfahrtsanzeige, nur mit folgenden Unterschieden:

- Statt der Abfahrtszeit wird die planmäßige Ankunftszeit angezeigt
- Die Auflistung in Spalte 3 soll statt der nachfolgenden die vorhergehenden Bahnhöfe mit deren fahrplanmäßiger Abfahrtszeit (SollZeit) anzeigen.

## Zuglauf

Ähnlich der Zuglaufdarstellung im DB Navigator soll der Zuglauf in Form einer senkrechten Perlenschnur mit dem Startbahnhof oben, dem Zielbahnhof unten und den Zwischenbahnhöfen als Perlen an der Schnur dargestellt werden. An jedem Bahnhof sollen folgende Informationen angezeigt werden:

1. Bahnstationsname
2. planmäßige Ankunftszeit und die Verspätungsminuten aus IstAnkunft - SollAnkunft
3. planmäßiges Bahnstationsgleis und in Klammern das tatsächlich durchfahrene Gleis (IstGleis)
4. planmäßige Abfahrtszeit und die Verspätungsminuten aus IstAbfahrt - SollAbfahrt
5. Übermittelter Meldungstext

Bei Bahnhöfen, die ohne Halte durchfahren werden, soll die Ankunftszeit und das Bahnstationsgleis nicht angezeigt werden.

In der Kopfzeile soll als seitenabhängiger Text die Gattung und die Nummer des Zuges angezeigt werden.

## Netzgrafik

Als extra Kundenwunsch soll während der Arbeit noch ein Featureswunsch eingefügt werden. Als Wunsch wird eine

Netzgrafik gewünscht, welche die Zugverbindungen innerhalb des Fahrplanes anzeigt. Diese soll auf Basis von per Telegramm gelieferten Betriebsstellenkoordinaten und den Zugverbindungen innerhalb des Fahrplans aufgebaut werden. Die Betriebsstellen sollen anklickbar sein und zu den jeweiligen Abfahrtsanzeigen führen. Nach Anklicken der Betriebsstelle sollen alle an von dieser Betriebsstelle ausgehenden Verbindungen hervorgehoben werden.

---

## Datenquellen

Das webbasierte Fahrgastinformationssystem soll im Endzustand auf einem Laborinternen Webserver laufen und seine Daten aus zwei Quellen bekommen:

- per Fahrplanserver über standardisierte TCP/IP Telegramme
- über Dateien vom Lokalen oder Netzlaufwerk.

### Dateien vom Lokalen oder Netzlaufwerk

Auf Dateiebene soll der das FIS folgende Dateien laden können:

- Konfigurationsdatei im INI-Format
- Liste von Meldungstexten mit deren Zuordnung im CSV-Format (Index;Meldung)
- XML-Fahrplandatei im RailML-Format
- etwaige Grafiken, die zur Darstellung nötig sind

Die Konfigurationsdatei sollte neben dem Programm auf dem gleichen Computer liegen. Diese soll auch die Pfade zu den beiden danach genannten Dateien beinhalten, da diese auf einem Netzlaufwerk liegen werden.

Die Liste der Meldungstexte ist nötig, da zum einsparen von Netzwerkverkehr statt der jeweiligen Meldung ein Meldungsindex verschickt wird. Um die zum Index zugehörige Meldung anzuzeigen, ist die Meldungstexte-Datei nötig.

Die XML-Fahrplandatei beinhaltet den geplanten Fahrplan ohne die Echtzeitinformationen. Diese soll beim Start des FIS einmalig geladen werden und die Daten in eine interne Datenstruktur überführt werden, wenn zum Startzeitpunkt keine Verbindung / Anmeldung am Fahrplanserver möglich ist. Dadurch wird sichergestellt, dass bei Abwesenheit des Fahrplanservers wenigstens Geplante Fahrplandaten angezeigt werden.

### Daten vom Fahrplanserver

Wenn das FIS sich am Fahrplanserver anmeldet, sendet der Fahrplanserver zuerst ein Bestätigungstelegramm mit der aktuellen Laboruhrzeit. Danach werden alle Zugläufe im Fahrplan mit ihren aktuellen Daten zum FIS gesendet. Danach beginnt der Fahrplanserver in unregelmäßigen Abständen Änderungstelegramme mit Echtzeitdaten an das FIS zu senden. Nebendessen wird auch regelmäßig die aktuelle Laborzeit vom Fahrplanserver verschickt, da diese unabhängig von der echten Uhr läuft.

Übersicht der Telegramme vom FIS zum Fahrplanserver:

- **Anmeldetelegramm** mit der Clientnummer nach dem Aufbau der TCP/IP-Verbindung
- **Clientstatustelegramm** mit der im Server anzuzeigenden Bezeichnung für den Client

Übersicht der Telegramme vom Fahrplanserver zum FIS:

- **Statustelegramm** mit der aktuellen Laborzeit als Bestätigung auf das Anmeldetelegramm
  - Regelmäßiges **Statustelegramm** mit der Laborzeit
  - **Zuglauftelegramm** mit den derzeitigen Informationen zu einem Zuglauf
  - Regelmäßige **Zuglauftelegramme** mit Echtzeitdaten
  - **Betriebsstellenbezeichnungstelegramm** am Anfang vor Übertragung der Zuglauftelegramme zur Dekodierung der verwendeten BahnhofIDs
-

Die Beschreibung der Telegramme und den Kommunikationsverlauf ist weiter unten, sowie unter Fahrplanserver und eFZM beschrieben.

## Weitere Anforderungen

Neben den im vorherigen Text implizit genannten Anforderungen werden hier noch weitere Anforderungen genannt, die das Programm erfüllen muss.

Anforderungskürzel	Beschreibung
W1	Wenn keine Verbindung zum Fahrplanserver besteht, soll die Uhr in der Kopfzeile ausgeblendet werden, da die Laboruhr anders als die Rechneruhr läuft.
W2	Alle derzeit geöffneten Seiten (Ankunfts-, Abfahrts- sowie Zuglaufseiten) sollen im Minutentakt aktuelle gehalten werden, um die Echtzeitinformationen sowie die Laborzeit in allen Ansichten aktuell zu halten. Wenn es eine andere Möglichkeit gibt die Anzeigen aktuell zu halten, dann kann diese Begründet stattdessen angewendet werden.
W3	Die Verbindungsstatusanzeige zum Fahrplanserver in der Fußleiste soll mindestens folgende Statusmeldungen anzeigen können: „Keine Verbindung“, „Verbindung wird aufgebaut...“, „Verbunden“.
W4	Das Logo in der Kopfzeile, sowie wenn weitere Grafiken verwendet werden, wird vom Datentyp Gif, Jpeg oder Png sein und im Programmverzeichnis liegen.
W5	Alle Farben in auf den Internetseiten sollen, soweit möglich, über die Konfigurationsdatei änderbar sein und es sollten nicht mehr als 6 Farben insgesamt verwendet werden.
W6	Die Vor- und Zurückfunktionen der Browser sollen verwendet werden können.
W7	Das FIS soll sich Innerhalb des Browsers auch in mehreren Tabs öffnen lassen.
W8	Die interne Speicherung soll nicht in einer Datenbank erfolgen.
W9	Die Bahnhöfe sollen intern in einer Klasse repräsentiert werden, damit zukünftig eine Erweiterung, um z.B. Umsteigeinformationen, leichter möglich ist.
W10	Die Seite an sich sollte auf Interaktionen mit dem Benutzer unmittelbar reagieren (z.B. Klick auf Link).
W11	Die Konfigurationsdatei soll mindestens folgende Informationen beinhalten: IP des Fahrplanservers, Port des Fahrplanservers, ClientID (mit der sich das FIS am Fahrplanserver anmeldet), Kennungsnummern der einzelnen Telegramme.

## Interne Datenstruktur

Der Aufbau der internen Datenstruktur können die Entwickler nach eigenem Ermessen machen. Als Empfehlung ist hier die derzeit im Labor genannte Klassenstruktur genannt:

### Klasse Zuglauf

- Zuggattung als String
- Zugnummer als Integer
- Bahnhoftsliste als Liste vom Typ Bahnhof

## Klasse Bahnhof

- Bahnhofskürzel als String
  - AnkunftSoll als DateTime
  - AbfahrtSoll als DateTime
  - GleisSoll als Byte
  - AnkunftIst als DateTime
  - AbfahrtIst als DateTime
  - GleisIst als Byte
  - Meldungstext als String
- 

## Umrechnung Uhrzeit

Ein besonderer Punkt ist die Umrechnung der per Telegramm gelieferten Uhrzeit. Um Speicherplatz zu sparen, ist im Labor die Uhrzeit in 2-Byte großen Datentyp Word als 10tel Minuten gespeichert. Die Zeit kann dort in 2 Zeitbereichen und bestimmten Werten gespeichert sein. Diese wären:

- 0 - 14400 → Aktueller Tag von 0 bis 24 Uhr
  - 20000 - 34400 → Nächster Tag von 0 bis 24 Uhr
  - 44444 → Zeit nicht relevant
  - 55555 → Durchfahrtszeichen
- 

## Telegramme

Hier wird zuerst der Grundlegende Telegrammaufbau, danach die benutzten Telegramme mit derer Abfolge aufgeführt.

### Grundlegender Telegrammaufbau

Ein Telegramm ist grundsätzlich wie folgt aufgebaut und maximal 255 Bytes lang:

1. **Startkennung** (Bytes [-3..-1]) mit jeweils 255 als Wert
2. **Längenangabe** angehängten, genutzten Nutzbytes (Byte [0])
3. Als erstes Nutzbyte eine **Kennung**, welche beschreibt, um was es sich für ein Telegramm handelt (Byte [1])
4. Als weiter Nutzbytes der jeweilige **Datensatz** (Record), der der Kennung entspricht (Byte [2..X], maximal bis 251)

Dieser Telegrammaufbau ist bei allen Telegrammen gleich. Es variiert nur die Anzahl der genutzten Nutzbytes. Die Länge der Nutzbytes ist immer 251 Bytes.

### Kommunikation Fahrplanserver <-> Fahrplanclients

Die Clients senden folgende Telegramme an den Fahrplanserver:

- Anmeldetelegramm mit der ClientID
- Clientstatustelegramm (einmalig nach Eintreffen des Zuglaufendetelegramms)

Der Server schickt folgende Telegramme an die Clients:

- Laborzeittelegramm (als Antwort auf die Anmeldung + Regelmäßig)
  - Zuglauftelegramme (nach Anmeldung über alle Züge und dann nur bei jeder Änderung)
  - Zuglaufendetelegramm (nachdem alle Zugläufe nach Anmeldung übertragen wurden)
  - Betriebsstellenbezeichnungstelegramm (nach Anmeldung und dann nach Anforderung durch den Client)
-

Telegrammname	Kennung	Inhalt mit Typ und Nr. in Bytearray
Anmeldetelegramm	251	1. ClientID [Byte] (224) 2. Laboruhr-Faktor [Byte] (default=1) 3. Züge Stopp/Weiter [Byte] (default=1)
Laborzeitletgramm	241	1. Stunde [Byte] 2. Minute [Byte] 3. Sekunde [Byte] 4. Laboruhr-Faktor [Byte] 5. Züge Stopp/Weiter [Byte]
Clientstatustelegramm	243	1.-7. Bezeichnung [Char] 8. Status [Byte]
Zuglaufetelegramm	236	1. FplNr [Byte] (0=Spieleise 0<FplNr) 2. Länge ZugNr String [Byte] 3-8. ZugNr [String] 9-13. Übergang in Zugnummer [Integer] 14. Länge Gattung String [Byte] 15-20. Gattung [String] 21. Tzf1 [Byte] 22-25. ZMA [Integer] (Zugmeldeabschnitt, in der der Zug gerade ist) 26. MldgIdx [Byte] (Index eines Meldetextes für den Zuglauf) 27. AnzBf [Byte] (Anzahl der nachfolgenden Betriebsstellen im Zuglauf)  - Ab hier bis zu 10 Betriebsstellen mit je folgender Datenstruktur - X. Bf_ID [Byte] (Nummer des Bahnhofs) X+1-2. AnkSoll [Word] X+3-4. AbfSoll [Word] X+5.-6. AnkVersp [SmallInt] (Verfrühung/Verspätung im Istzustand) X+7.-8. AbfVersp [SmallInt] (Verfrühung/Verspätung im Istzustand) X+9. GleisSoll [Byte] X+10. GleisDispo [Byte] X+11. DispoTyp [Byte] (Index der anzuzeigenden DispoBilder) X+12. MldgIdx[Byte] (Index eines Betriebsstellenbezogenen Meldungstextes)
Zuglaufendetelegramm	236	1. Zahl 255 [Byte]
BS Bezeichnungstelegramm	238	1. Betriebsstellen ID [Byte] (Zuordnungsnummer des Betriebsstellennamens zu den Bf_ID in den Zugläufen) 2. Länge Abkürzung String [Byte] 3.-7. Abkürzung [String] (Verkürzte Bezeichnung) 8.- X-8. Betriebsstellenlangname [String] (Langbezeichnung) X-7. - X-4. Koordinate X [Single/Float] (ist 4 Byte) X-3. - X. Koordinate Y [Single/Float] (ist 4 Byte)

#### Bemerkungen zu den Telegrammen:

- Zum Verringern der Telegrammgröße wird statt des Betriebsstellenkürzels die Betriebsstellen Nummer übertragen. Zur Zuordnung der Betriebsstellennummer zu einem Bahnhofsnamen werden nach dem Laborzeitletgramm, als Anmeldebestätigung, nacheinander alle benötigten Nummer↔Name Zuordnungen als Betriebsstellenbezeichnungstelegramme verschickt. Danach werden alle Zuglaufetelegramme mit dem Zuglaufendetelegramm am Ende verschickt.
- Nach dem Zuglaufendetelegramm kann einmalig das Clientstatustelegramm verschickt werden, mit einer vom Programmierer frei wählbaren maximal 6stelligen Bezeichnung des Clients. Im aktuellen Fall z.B. „FIS“.
- Nach der Zugnummer können bis zu 10 Betriebsstellen kommen. Die Anzahl der Betriebsstellen steht immer im

AnzBf Byte.

- Die Soll Ab-/An-Zeiten, sowie die Verspätungen in 10tel Minuten übertragen. Die Ist-Zeiten werden durch Addieren der Verspätungen zu den Zeiten errechnet. Errechnet werden die einzelnen Zeitkomponenten wie folgt: 10tel Minute:  $\text{Zeit} \bmod 10$ ; Minute:  $(\text{Zeit}/10) \bmod 60$ ; Stunde:  $\text{Zeit}/600$ .
- Die Versendeten Strings und Chars sind in ANSI Codierung.

### Kommunikation nach dem Anmeldevorgang

Nach dem Aufbau der TCP/IP-Verbindung verschickt und empfängt das FIS diverse Telegramme. Eine beispielhafte Kommunikation ist nachfolgend abgebildet.

ausgehend ←	Anmeldetelegramm (Kennung 251): AnmeldeID=41; tEBL= 1; ZugStatus= 1;
eingehend →	Laborzeittelegramm (Kennung 241): 11; 45; 26; 60; 2;
eingehend →	BS Bezeichnungstelegramm (Kennung 238): ... n Telegramme ...
eingehend →	Zuglauftelegramme (Kennung 236): ... n Telegramme ...
eingehend →	Zuglaufendetelegramme (Kennung 236): 255
ausgehend ←	Clientstatustelegramm (Kennung 243): Bezeichnung=FIS; Status=0

## Abgabe

Am Ende des Softwarepraktikums soll für das Labor folgendes abgegeben werden:

- Das lauffähige Webprogramm
- Der Quellcode
- Eine Entwicklerdokumentation mit dem Aufbau des Codes und mindestens folgender Beschreibungen:
  - an welchen Stellen im Code die Ausgaben und das Design der Oberfläche modifiziert werden kann
  - an welchen Stellen im Code die Telegramme verarbeitet werden
  - Aufbau und Einleseort der Configurationsdatei
  - Unter welchen Voraussetzungen das Webprogramm stabil läuft

Weitere Vorgaben werden von der Informatik bekanntgegeben.