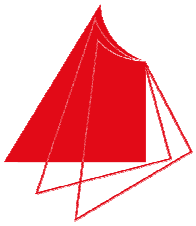

Hochschule Karlsruhe
Technik und Wirtschaft
UNIVERSITY OF APPLIED SCIENCES

BMTIS:

Bostoner Metro- und Touristeninformationssystem

Studiengang Kartographie und Geomatik
Praktikum Datenbanken und Informationssysteme
WS 2008 / 2009
Dozent: Prof. H. Kern
Bearbeitet von Ralf Seger

2. Aufgabenstellung

Hochschule Karlsruhe - Technik und Wirtschaft
Fakultät für Geomatik
Studiengang Kartographie und Geomatik
Prof. Hans Kern

den 6. Oktober 2008

Name.....

PDB K6B42, K6D15, K5D42 und K6D62

Der Leistungsnachweis zu diesem Praktikum besteht für K6B42 (PO 1), K6D15 (PO 2) und K6D62 (PO 4) aus einer nicht benoteten, für K6D62 (PO 4) aus einer benoteten Studienarbeit, die in Form einer Projektarbeit zu erbringen ist. Das Projekt verfolgt einen integrativen Ansatz, indem die Inhalte vorangegangener Lehrveranstaltungen wie Datenbanken und Informationssysteme, Betriebssysteme, Programmieren, Integrale Kartographie und Mediendesign und Integration zur Lösung der Projektaufgaben herangezogen werden.
Als Grundlage dienen die Daten vorangegangener Projekte. Es handelt sich dabei um Verkehrsnetze größerer Städte mit zusätzlichen touristischen Informationen.

Aufgaben

--Sichtung und Bewertung der vorhandenen Daten (Projektanalyse). Sie erhalten die Daten eines vor
angegangenen Projekts und eine Mustervorlage für Ihr eigenes Projekt.
--Ergänzung und/oder Vervollständigung der vorhandenen Daten. Es müssen zusätzliche Daten mit geographischer Referenz in Form geographischer Koordinaten erfaßt werden; z.B. Religiöse Stätten, Preiszonen, Restaurants, Fahrpläne. Falls die Koordinaten der Vorlage keine Gauß-Krüger- oder geographischen Koordinaten sind, reicht auch die Umwandlung in solche.
--Erstellen einer SQL-Ladedatei, die sämtliche Informationen bereitstellt. Laden der Daten in eine MySQL-Datenbank.
--Installation der Software. Sie brauchen Notepad++, Apache, MySQL und PHP, Perl, JSP oder ASP. Wenn Sie XAMPP installieren, stehen Ihnen Apache, MySQL, PHP und Perl zur Verfügung.
--Auffrischung der Kenntnisse in HTML; insbesondere Formulare (Checkbutton, Radiobutton, Auswahlliste, Textfeld, Schaltfläche) und Cascading Style Sheets.
--Konzeption des Layouts für die Site mit CSS oder Frames.
--Auffrischung der Kenntnisse in einer serverseitigen Skriptsprache. Sie können PHP, Perl, JSP oder ASP verwenden. Sie sollten die Kommunikation zwischen Browser und Server bei der Verwendung von Formularen testen. Danach sollten Sie die Anbindung an eine Datenbank versuchen.
--Realisierung von 20 Abfragen an die Datenbank. Es sollen alle Formularelemente benutzt werden und Entfernungs- und Erreichbarkeitsabfragen enthalten sein.
--Interaktive Darstellung von Punkt- und möglichst auch Linienobjekten mit GoogleMaps.

Anregung

Dieses Aufgabenblatt ist Bestandteil der Studienarbeit. Sie sollen Ihr Projekt dokumentieren und erläutern. Bei den einzelnen Abfragen können Sie dann auf die inhaltlichen und technischen Aspekte eingehen. Die Eingaben und Ausgaben sollen vollständig dokumentiert sein. Die Zuordnung von Ausgabe zu Eingabe soll leicht erkennbar sein. EDV-Ein- und Ausgaben sind in Courier wiederzugeben.

Hochschule Karlsruhe - Technik und Wirtschaft
Fakultät für Geomatik
Studiengang Kartographie und Geomatik
Prof. Hans Kern

den 6. Oktober 2008

Name.....

Bitte beachten Sie die Hinweise in der Arbeitsanleitung.
Für die Bewertung sind die Vollständigkeit der Dokumentation, die technische Komplexität, sowie die Gestaltung der Unterlagen wichtig (CD und Druckausgabe im Hochformat). Dokumentieren Sie differenziert Ihren Zeitbedarf. Arbeit in Gruppen von höchstens zwei Personen. Bitte keine Klarsichthüllen verwenden.

Termine

Einführung	10.10.2008
Vorführung am Rechner	ab 16.1.2009
Abgabe der Studienarbeit als Ausdruck und auf CD	spätestens am 30.1.2009

Literatur

Stefan Hinz u. Michael Seeboerger-Weichselbaum, MySQL 5 ge-packt, Redline, 2006
Wolfgang D. Misgeld, ORACLE für Profis; München, Wien; Hanser 1991
Eva Kraut u. Theodor Seidl: Oracle-SQL, it-Verlag; o.J.
Gregor Kuhlmann u. Friedrich Müllmerstaft: SQL Der Schlüssel zu rel. Datenbanken, Rowohlt, 2000
Alex Morrison u. Alice Rischert: Oracle SQL, Prentice Hall, ISBN 0-13-015745-7
Stefan Mintert (Hg.): XHTML, CSS & Co, Addison-Wesley, 2003
Chuck Musciano u. Bill Kennedey: HTML-Das umfassende Referenzwerk, O´Reilly, 1997
Rasmus Lerdorf u. Kevin Tatroe: Programmieren mit PHP, O´Reilly, 2003
Rasmus Lerdorf: PHP kurz und gut, O´Reilly, 2003

3. Projektanalyse

3.1 Datengrundlage

Grundlage dieser Arbeit ist die Studienarbeit von Sarah Ostheimer aus dem Wintersemester 2006/ 2007.

An Daten standen zur Verfügung:

- fünf Tabellen im CSV- und XLS-Format
- 13 Datenbankabfragen im TXT-Format
- die Dokumentation der Autorin
- ein Netzplan im PNG-Format

3.2 Prüfung und Ergänzung der Daten

Die Daten in der Datenbank waren zum Teil fehlerhaft. Speziell die Tabelle namens Haltestellen_Linien musste überarbeitet werden, da der Verlauf der Linien nicht konsistent, sondern lückenhaft war. So waren zum Beispiel bei vier von acht Linien Lücken im Linienverlauf zu finden, die eine direkte Weiterwendung unmöglich machten. Die fehlenden Haltestellen, sowie die dazugehörenden Daten wie die Reihenfolge, Fahrtdauer und Distanz zwischen den Haltestellen mussten nachrecherchiert und eingepflegt werden.

Die Datenbankabfragen in der Textdatei waren inhaltlich nur bedingt zu gebrauchen, weil sie ausschließlich statisch und für die Ausgabe in der Konsole konzipiert waren. Dabei sollten die Anfragen der aktuellen Aufgabenstellung nach doch dynamisch und wesentlich komplexer sein, um auf HTML-Seiten mit Google Maps eingebunden werden zu können. Sie dienten deshalb nur der Auffrischung der SQL -Kenntnisse und als Inspiration für komplexere Abfragen.

Die Dokumentation dagegen war ausführlich und half sehr beim Kennenlernen der vorhandenen Datenbankstruktur.

3.3 Arbeitsschritte

Folgende Arbeitsschritte wurden im Vorfeld zur Bearbeitung des Projektes angedacht und auch in etwa in dieser Reihenfolge abgearbeitet:

1) Installation der notwendigen Software und Einrichtung der Arbeitsumgebung
2) Sichten und Korrigieren der Datenbankeinträge
3) Einarbeitung in SQL, Studium der SQL-Abfragen von Frau Ostheimer

4) Formulieren und testen eigener SQL-Abfragen
5) Erarbeiten eines Konzeptes zum Layout der HTML-Seite
6) Einarbeiten in CSS um ein HTML-Grundgerüst zu entwickeln
7) Studium von PHP, was die HTML-Seiten dynamisch erzeugen sollte
8) Programmierung der PHP-Seiten mit Implementierung der SQL-Abfragen
9) Testen und Fehlersuche
10) Dokumentation erstellen.

4. Die Datenbank hinter BMTIS

4.1 Einführung

Die Datenbank hinter BMTIS besteht aus fünf Tabellen:
Die erste Tabelle „track“ beinhaltet Namen und Farbtöne der U-Bahn-Linien von Boston. In der zweiten Tabelle „station“ sind alle Haltesstationen mit Namen und ihren Koordinaten gespeichert. Um die Daten dieser beiden Tabellen zu verknüpfen, gibt es die Tabelle „s_t“, was für „station_track“ steht. In ihr sind zusätzlich Daten über die Reihenfolge der Haltestellen, sowie die räumliche und zeitliche Entfernung hinterlegt. Die Tabellen „poi“ und „typ“ enthalten Daten über die „Points of Interest“, also über interessante Gebäude oder Sehenswürdigkeiten, für die sich Touristen beim Aufenthalt in Boston interessieren könnten. In der Tabelle „poi“ findet man die Namen der „Points of Interest“, deren Koordinaten und eine ausführliche Beschreibung. Die Tabelle „typ“ enthält lediglich die Klassifizierung der Sehenswürdigkeiten zum Beispiel in Museum oder Übernachtungsmöglichkeiten.

Zusätzlich haben alle Tabellen außer „s_t“ eine Spalte mit der „ID“ als Primärschlüssel. In der Tabelle „s_t“ werden die „ID-Schlüssel“ von „track“ und „station“ als Fremdschlüssel verwendet, um die oben beschriebene Verknüpfung zu ermöglichen.
In der Tabelle „poi“ befindet sich ebenfalls ein Fremdschlüssel „type_id“, der dem Primärschlüssel der Tabelle „typ“ (Kategorisierung der poi) entspricht.

4.2 Entity-Relationship-Modell der Datenbank

Das folgende ER – Modell wurde von Frau Ostheimer übernommen und überarbeitet, da die Bezeichnungen der Tabellenspalten in Englisch geändert wurden.

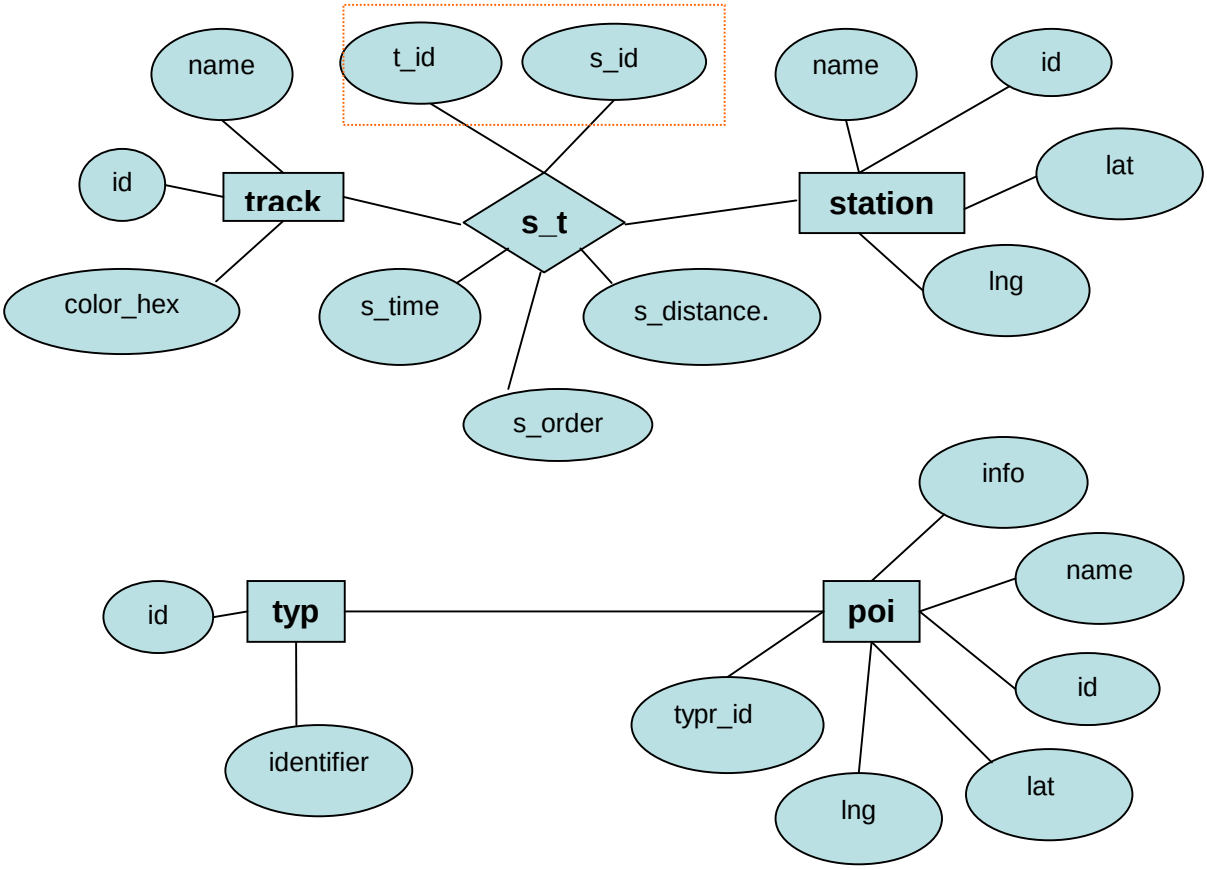

Abb. 1: Entity-Relationship-Diagramm der Datenbank boston

4.3 Anlegen der Datenbank

Die Datenbank wurde mittels einer Ladedatei erzeugt, mit der sämtliche Tabellen erst erzeugt und dann mit den Daten befüllt wurden. Dies geschah in einem Schritt, da der Gesamtinhalt als ein einziges SQL-Query formuliert wurde. Auf den folgenden Seiten werden der Übersicht wegen die einzelnen Befehlsgruppen differenziert dargestellt.

4.3.1 Tabelle „track“:

```
CREATE TABLE track (
     id Int (3) NOT NULL auto_increment,
     name VARCHAR (100) NOT NULL,
     color_hex varchar (7) NOT NULL,
     PRIMARY KEY (Id)
     );

INSERT INTO track (id,name,color_hex) VALUES
(1,'Red Line A','#FA2D27'),
(2,'Red Line B','#FA2D27'),
(3,'Orange Line','#FD8A03'),
```

```
(4,'Green Line B','#008150'),
(5,'Green Line C','#008150'),
(6,'Green Line D','#008150'),
(7,'Green Line E','#008150'),
(8,'Blue Line','#2F5DA6');
```

4.3.2 Tabelle „station“:

```
CREATE TABLE station (
     id INT (3) NOT NULL auto_increment,
     name VARCHAR (50) NOT NULL,
     lat FLOAT NOT NULL,
     lng FLOAT NOT NULL,
     PRIMARY KEY (Id)
     );

INSERT INTO station (id,name,lat,lng) VALUES
(1,'Bowdoin',42.361457,-71.062129),
(2,'Government Center',42.359322,-71.059252),
(3,'State',42.359065,-71.057421),
(4,'Aquarium',42.359634,-71.051807),
(5,'Maverick',42.368343,-71.038422),
(6,'Logan Airport',42.372714,-71.03208),
(7,'Wood Island',42.379759,-71.022476),
(8,'Orient Hights',42.386983,-71.004767),
(9,'Suffolk Downs',42.389899,-70.997986),
(10,'Beachmont',42.39745,-70.992363),
(11,'Revere Beach',42.408389,-70.99256),
(12,'Wonderland',42.413963,-70.990986),
(13,'Forest Hills',42.300023,-71.113377),
(14,'Green Street',42.310359,-71.107125),
(15,'Stony Brook',42.317251,-71.104021),
(16,'Jackson Square',42.322893,-71.099787),
(17,'Roxbury Crossing',42.331388,-71.095555),
(18,'Ruggles',42.33572,-71.090437),
(19,'Massachusetts Ave',42.341762,-71.083072),
(20,'Back Bay',42.341762,-71.083072),
(21,'New England',42.341762,-71.083072),
(22,'Chinatown',42.352529,-71.062759),
(23,'Downtown Crossing',42.355453,-71.060465),
(24,'Haymarket',42.363222,-71.057922),
(25,'North Station',42.365551,-71.061251),
(26,'Community College',42.372593,-71.07052),
(27,'Sullivan Square',42.383434,-71.077056),
(28,'Wellington',42.405202,-71.076969),
(29,'Malden Center',42.426678,-71.074144),
(30,'Oak Grove',42.437134,-71.070888),
(31,'Braintree',42.20855,-71.00085),
(32,'Quincy Adams',42.232848,-71.007034),
(33,'Quincy Center',42.250879,-71.004798),
(34,'Wollaston',42.265972,-71.019721),
```

```
(35,'North Quincy',42.274957,-71.029307),
(36,'JFK',42.321065,-71.052545),
(37,'Andrew',42.329752,-71.056979),
(38,'Broadway',42.342793,-71.057117),
(39,'South Station',42.352573,-71.055428),
(40,'Park Street',42.356332,-71.062202),
(41,'Charles MGH',42.361279,-71.070493),
(42,'Kendall',42.362427,-71.086058),
(43,'Central',42.365171,-71.103386),
(44,'Harvard',42.373936,-71.118917),
(45,'Porter',42.388353,-71.119159),
(46,'Davis ',42.39662,-71.122527),
(47,'Alewife',42.395261,-71.14244),
(48,'Mattapan',42.267586,-71.092021),
(49,'Capen St',42.267622,-71.087436),
(50,'Valley Rd',42.267772,-71.083025),
(51,'Central Ave',42.269965,-71.073249),
(52,'Milton',42.270093,-71.067612),
(53,'Butler',42.272253,-71.062453),
(54,'Cedar Grove',42.279712,-71.060327),
(55,'Ashmont',42.284219,-71.063229),
(56,'Shawmut',42.293712,-71.065912),
(57,'Fields Corner',42.299992,-71.061516),
(58,'Savin Hill',42.311099,-71.053175),
(59,'Heath',42.32868,-71.11068),
(60,'Back of the Hill',42.32996,-71.111406),
(61,'Riverway',42.331686,-71.112004),
(62,'Mission Park',42.333106,-71.109678),
(63,'Fenwood Road',42.333722,-71.105682),
(64,'Brigham Circle',42.334268,-71.10437),
(65,'Longwood Medical Area',42.335898,-71.100052),
(66,'Fine Museum of Arts',42.337732,-71.095125),
(67,'Northeastem',42.340307,-71.088717),
(68,'Symphony',42.342919,-71.084529),
(69,'Prudential',42.345503,-71.081401),
(70,'Copley',42.349991,-71.077463),
(71,'Arlington',42.351847,-71.070817),
(72,'Boylston',42.352337,-71.06461),
(73,'Science Park',42.366725,-71.067681),
(74,'Lechmere',42.370774,-71.076593),
(75,'Riverside',42.337394,-71.252327),
(76,'Woodland',42.333336,-71.24434),
(77,'Waban',42.325981,-71.230712),
(78,'Elliot',42.318744,-71.217023),
(79,'Newton Highlands',42.321539,-71.205966),
(80,'Newton Centre',42.328354,-71.194444),
(81,'Chestnot Hill',42.326601,-71.16529),
(82,'Reservoir',42.334912,-71.149072),
(83,'Beaconsfield',42.335805,-71.140849),
(84,'Brookline Hills',42.331133,-71.127031),
(85,'Brookline Village',42.332566,-71.116948),
(86,'Longwood',42.341042,-71.110215),
```

```
(87,'Fenway',42.345347,-71.104156),
(88,'Cleveland Circle',42.336197,-71.149406),
(89,'Englewood Ave',42.337049,-71.145357),
(90,'Dean Road',42.337847,-71.141549),
(91,'Tappan Street',42.338469,-71.138705),
(92,'Washington Square',42.339258,-71.135387),
(93,'Fairbank Street',42.339474,-71.131304),
(94,'Brandon Hall',42.339683,-71.129327),
(96,'Coolidge Corner',42.342226,-71.120888),
(97,'St Paul Street',42.343366,-71.116759),
(98,'Kent Street',42.344125,-71.113885),
(99,'Hawes Street',42.34487,-71.111129),
(100,'St Mary Street',42.345947,-71.107385),
(101,'Boston College',42.339902,-71.166124),
(103,'South Street',42.339455,-71.157622),
(104,'Chestnut Hill Avenue',42.338145,-71.152961),
(105,'Chiswick Road',42.340839,-71.150459),
(106,'Sutherland Street',42.341483,-71.146162),
(108,'Washington Street',42.343886,-71.142593),
(109,'Summit Avenue',42.345806,-71.141232),
(110,'Warren Street',42.348366,-71.140224),
(111,'Allston street',42.348468,-71.137888),
(112,'Griggs Street',42.348542,-71.134551),
(113,'Harvard Ave',42.350118,-71.131197),
(115,'Packard Corner',42.351833,-71.124702),
(116,'Babcock Street',42.35169,-71.121547),
(117,'Pleasent Street',42.351345,-71.118783),
(118,'St Paul Street',42.350997,-71.115997),
(119,'Boston University West',42.350759,-71.113833),
(120,'Boston University Central',42.34989,-71.106804),
(121,'Boston University East',42.349569,-71.103866),
(122,'Bladford Street',42.349126,-71.100235),
(123,'Kenmore',42.348783,-71.095128),
(124,'Hynes Convention Centre',42.347745,-71.087744);
```

4.3.3 Tabelle „s_t“:

```
CREATE TABLE s_t (
     s_id INT (3) NOT NULL,
     t_id INT (3) NOT NULL,
     s_order INT (3) NOT NULL,
     s_time INT (3) NOT NULL,
     s_distance INT (6) NOT NULL,
     PRIMARY KEY (s_id, t_id));

INSERT INTO s_t (s_id,t_id,s_order,s_time,s_distance) VALUES
(1,8,1,0,0),
(2,8,2,1,400),
(3,8,3,2,660),
(4,8,4,4,1176),
```

```
(5,8,5,6,2591),
(6,8,6,7,3318),
(7,8,7,8,4273),
(8,8,8,10,6073),
(9,8,9,11,6741),
(10,8,10,12,7687),
(11,8,11,12,8848),
(12,8,12,15,9588),
(13,3,1,0,0),
(14,3,2,2,1398),
(15,3,3,4,2174),
(16,3,4,5,2954),
(17,3,5,6,3940),
(18,3,6,7,4708),
(19,3,7,8,5485),
(20,3,8,9,6415),
(21,3,9,10,7382),
(22,3,10,11,7808),
(23,3,11,12,8287),
(3,3,12,12,8691),
(24,3,13,14,9121),
(25,3,14,15,9476),
(26,3,15,16,10823),
(27,3,16,18,12017),
(28,3,17,20,14085),
(29,3,18,22,16742),
(30,3,19,21,17936),
(31,1,1,0,0),
(32,1,2,4,3069),
(33,1,3,6,5017),
(34,1,4,8,7039),
(35,1,5,10,8335),
(36,1,6,16,14169),
(37,1,7,18,15389),
(38,1,8,19,16798),
(39,1,9,20,17895),
(23,1,10,21,18648),
(40,1,11,22,18266),
(41,1,12,23,19197),
(42,1,13,24,20235),
(43,1,14,26,21962),
(44,1,15,28,23502),
(45,1,16,30,25026),
(46,1,17,31,26161),
(47,1,18,32,27113),
(48,2,1,0,0),
(49,2,2,1,518),
(50,2,3,2,914),
(51,2,4,3,1652),
(52,2,5,4,2178),
(53,2,6,5,2576),
(54,2,7,6,3442),
```

```
(55,2,8,7,4204),
(56,2,9,8,5166),
(57,2,10,9,6193),
(58,2,11,11,7764),
(36,2,12,14,8321),
(59,7,1,0,0),
(60,7,2,1,154),
(61,7,3,2,302),
(62,7,4,3,626),
(63,7,5,4,925),
(64,7,6,5,1027),
(65,7,7,6,1432),
(66,7,8,7,1914),
(67,7,9,8,2626),
(68,7,10,9,2905),
(69,7,11,10,3368),
(70,7,12,11,3955),
(71,7,13,12,4550),
(72,7,14,13,5038),
(40,7,15,14,5518),
(2,7,16,15,5935),
(24,7,17,16,6382),
(25,7,18,17,6738),
(73,7,19,18,7428),
(74,7,20,19,8321),
(75,6,1,0,0),
(76,6,2,1,663),
(77,6,3,3,2051),
(78,6,4,4,3511),
(79,6,5,5,4635),
(80,6,6,7,6038),
(81,6,7,8,8355),
(82,6,8,9,9958),
(83,6,9,10,10709),
(84,6,10,11,11933),
(85,6,11,12,12759),
(86,6,12,13,13835),
(87,6,13,14,14500),
(123,6,14,15,15676),
(88,5,1,0,0),
(89,5,2,1,501),
(90,5,3,2,729),
(91,5,4,3,962),
(92,5,5,4,1260),
(93,5,6,5,1650),
(94,5,7,6,1840),
(95,5,8,7,2214),
(96,5,9,8,2478),
(97,5,10,9,2832),
(98,5,11,10,3081),
(99,5,12,11,3389),
(100,5,13,12,3761),
```

```
(123,5,14,13,3855),
(101,4,1,0,0),
(102,4,2,1,488),
(103,4,3,2,804),
(104,4,4,3,1200),
(105,4,5,4,1493),
(106,4,6,5,1957),
(107,4,7,6,2140),
(108,4,8,7,2353),
(109,4,9,8,2601),
(110,4,10,9,2926),
(111,4,11,10,3133),
(112,4,12,11,3386),
(113,4,13,12,3758),
(114,4,14,13,3998),
(115,4,15,14,4310),
(116,4,16,15,4635),
(117,4,17,16,4872),
(118,4,18,17,5090),
(119,4,19,18,5297),
(120,4,20,19,5843),
(121,4,21,20,6029),
(122,4,22,21,6364),
(123,4,23,22,6789),
(124,4,24,23,7440),
(70,4,25,24,7784);
```

4.3.4 Tabelle „poi“:

```
CREATE TABLE poi (
     id INT (3) NOT NULL auto_increment,
     type_id INT (3) NOT NULL,
     name VARCHAR (100) NOT NULL,
     lat FLOAT NOT NULL,
     lng FLOAT NOT NULL,
     info VARCHAR (150) NOT NULL,
     PRIMARY KEY (Id)
     );


INSERT INTO poi (id,type_id,name,lat,lng,info) VALUES
(1,3,'Boston Charles Street Inn',42.358513,-71.070912,'Bostons
bestes bed and breakfast hotel in historischem Stil und in
angenehmer Atmosphäre.'),
(2,3,'Bulfinch Hotel',42.363697,-71.062132,'Ein typisches
Dreiecksgebäude im Herzen Bostons aus dem Jahre 1800. Es
enthaelt 80 Zimmer und Suiten.'),
(3,3,'Doubletree Hotel Boston',42.348808,-
71.064686,'Luxuriöses Hotel im Herzen Bostons. Beeinflusst
durch die Lage in Chinatown in edlem Feng-Shui Design.'),
```

```
(4,3,'The Copley House',42.344562,-71.080229,'Liegt im
kulturellen Center der Stadt, bestehend aus einzelnen, sehr
gut ausgestatteten Apartments und Studios'),
(5,3,'Coolidge Guest House',42.339479,-71.11912,'10 min.
außerhalb von Boston Downton .Edler viktorianischer
Architekturstil. Bekannt für Bagels. Low Budget.'),
(6,3,'Cambridge House', 42.393233,-71.126161,'Ein im typischen
New England Stil des letzen Jahrhunderts eingerichtetes Hause
nahe der Universitäten gelegen'),
(7,2,'New England Sports Museum',42.366234,-
71.061149,'Beinhaltet eine große Sammlung aus historischen
Sportereignissen New Englands. Bsp.: Hockey, Baseball,
Football.'),
(8,2,'The Computer Museum',42.368333,-71.071601,'Das heute zum
Museum of Science gehörende Gebäude beinhaltet neueste
Computer sowie Internet Technologien'),
(9,2,'MIT Museum',42.343957,-71.08613,'Ständig wechselnde
Ausstellungen zum Thema Technik und Wissenschaft, Architektur,
Design und Ozeanography.'),
(10,2,'Museum of Fine Arts',42.339408,-71.094214,'Beinhaltet
eine der größten Kunstsammlungen der Welt :3000 jahre alte
Raritäten; zeitgen. Kunst und Kunst von morgen!'),
(11,2,'Museum of Science',42.368333,-71.071601,'Wechselnde
Ausstellungen zu aktuellen wissenschaftlichen Themen. Große
Bekanntheit durch die Ausstellung Körperwelten'),
(12,1,'Opera House',42.292754,-71.071806,'Neu renoviert.Es
bietet eine opulente Alternative zu den zahlreichen im
benachbarten Theaterbezirk gelegenen Theatern.'),
(13,1,'Boston Public Gardens',42.354084,-
71.069924,'Bereits1634 gegründet zählt der Park zu DER
Grünfläche im dichtbesiedelten Downtown Center Bostons'),
(14,1,'John Hancock Tower',42.348519,-71.075372,'Das höchste
Gebäude New Englands von Stararchitekt Mies van der Rohe
beeindruckt durch seine Glasfasade und Design'),
(15,1,'Boston Symphony Hall',42.342843,-71.085689,'Erbaut in
1900 zählt es zu den 3 edelsten Konzerthallen der Welt. Im
Wechsel gibt es klassische Opernaufführungen und Pop.'),
(16,1,'Matthews Arena',42.341197,-71.084589,'Älteste
Eissporthalle der Welt und Heim der Boston Briuns.
Gelegentlich zum Schlittschulaufen geöffnet.'),
(17,1,'Mystic River Reservation',42.413164,-71.098039,'Freier
Eintritt. Der Park bietet Freizeitmöglichkeiten wie Tennis,
Beachball, Bootfahrten, Klettern, Radfahren, Segeln.'),
(18,1,'New England Aquarium',42.358986,-71.050666,'Größtes
Aquarium New Englands. Beinhaltet eine große Vielzahl an
Meeresbewohnern. Seit 2003 auch ein IMAX Kino'),
(19,1,'Playground',42.501467,-71.133697,'Eine Riesenhalle mit
umfangreichen Freizeitprogramm fuer Kinder ab 3 Jahren.
Familientageskarten stark verbilligt.'),
(20,1,'Paul Revers House',42.360391,-71.056618,'Histor.
Sehenswürdigkeit: Boston ältestes Gebaeude und eines der
wenigen noch erhaltenen aus der Kolonialzeit Amerikas'),
```

```
(21,1,'JFK Presidential Library',42.316752,-
71.058691,'Bibliothek und auch Museeum das historische
Ausstellungsstücke des 35. Präsidenten Amerikas vereint.'),
(22,1,'Captain Forbes House',42.262246,-71.060519,'Ausstellung
über Captain Forbes, der Bostons Wirtschaft zu einem enormen
Aufschwung in den 1830 er Jahren verhalf'),
(23,1,'Old South Meeting House',42.357005,-71.058724,'Das
Gebäude war zentraler Punkt der Boston Tea Party von 1773.
Heute finden hier Tagungen statt. Teilweise geöffnet'),
(24,1,'The Wang Center',42.350281,-71.06514,'Center für
Konferenzen, Filmvorführungen, Ausstellungen. Wunderschöne
Architektur und Gartenanlage im asiatischen Stil.'),
(25,1,'Old North Church',42.366529,-71.054474,'Offiziell als
Christus Kirche anerkannt und 1723 erbaut. Georgianischer
Architekturstil.Enthält Amerikas älteste Glocke'),
(26,1,'Kings Chapel',42.358003,-71.059981,'1668 als erste
nicht puritanische Kirche erbaut. Für die damalige Zeit
hervoragende Qualität der Orgel.'),
(27,1,'Freedom Trail',42.353008,-71.060816,'Ein 2.5 Meilen
langer Pfad, der zu 16 original nachgebauten Schauplätzen der
Amerikanischen Revolution führt.'),
(28,1,'Harvard University',42.374554,-71.116818,'1636
gegründet gilt sie als die älteste private Universität
Amerikas. Amerikanische Eliteuni.'),
(29,1,'Bunker Hill Monument',42.375587,-71.061418,'Der 67 m
hohe Granitobelisk soll an die Schlacht von Bunker Hill am 17.
Juni 1775 erinnern. Eine Wendeltreppe führt nach oben'),
(30,1,'Old Granary Burying Ground',42.357354,-
71.061778,'Friedhof berühmter Bostoner Persönlichkeiten mit
jahrhundertealten Grabsteinen'),
(31,1,'Orphium Theatre',42.356371,-71.061142,'Historisches
Gebäude von 1852. Tchaikovsky spielte hier sein 1. Piano
Concert. Heute spielen aktuelle Bands wie Coldplay'),
(32,4,'Boston Prudential Center',42.348691,-71.082523,'New
Englands Nummer 1 Shopping Center hat mehr als 75 Shops mit
edleren Marken : Lacoste, Sephora .Und einige Bars'),
(33,4,'Adidas Retail 504',42.372754,-71.116587,'Riesenauswahl
an Schuhen und Bekleidung der Marke Adidas zu günstigen
Preisen'),
(34,4,'Macys',42.355201,-71.060797,'Bekanntes Bekleidungshaus
mit einer Vielzahl an bekannten amerikanischen Fashion-
Marken.'),
(35,5,'Mc Ganns Pub',42.364513,-71.061709,'Irisher Pub in
rustikaler Atmosphäre'),
(36,5,'Mojitos',42.355963,-71.062025,'In dem Club wird
südamerikanische Musik gespielt. Bekannt für seine
extravagante Innenaustattung'),
(37,5,'All Asia Cuisine',42.362575,-71.098931,'Die Bar bietet
asiatische Spezialitäten und eine große Palette an Cocktails
an. August geschlossen.'),
(38,5,'Club 58',42.247774,-71.001346,'Angesagter Club der
Stadt . High Budget. Etwas außerhalb des Zentrums gelegen'),
```

```
(39,5,'Allston Rocks',42.35743,-71.129998,'Livekonzerte mit
Musikern aus der Umgebung. Musikrichtung ist Indie Rock. Einer
der Szenetreffs in Boston'),
(40,5,'Alley Cat Lounge',42.35215,-71.066102,'Centraler Club .
Ständig wechselndes Programm. After- Work Partys sind
empfehlenswert'),
(41,6,'Aquitaine',42.343613,-71.072295,'Französische
Spezialitäten. Gehobene Preisklasse'),
(42,6,'Umberto Galleria',42.363792,-71.054503,'Italienisches
Feinschmecker Restaurant. Gehobene Preisklasse'),
(43,6,'Mc Cornick and Schmick',42.359751,-71.05364,'Beliebtes
Restaurant. Großes Angebot an Fischgerichten und
Meeresfrüchten. Relativ günstig. Bitte vorher reservieren.'),
(44,6,'Ginza Japanese Restaurant',41.240099,-
81.451065,'Restaurant mit japanischen Gerichten');
```

4.3.5 Tabelle „typ“:

```
CREATE TABLE typ (
     id INT (3) NOT NULL auto_increment,
     identifier VARCHAR (50) NOT NULL,
     PRIMARY KEY (Id)
     );

INSERT INTO typ (id,identifier) VALUES
(1,'Sehenswördigkeit'),
(2,'Museum'),
(3,'Übernachtung'),
(4,'Shopping'),
(5,'Nightlife'),
(6,'Essen');
```

5. Gestaltung und Funktionsweise der Webseite

5.1 HTML und CSS

Im Gesamten gibt 4 HTM-Dateien, die das Grundgerüst der Webseite bilden. Sie sind im Folgenden:

- impressum.htm (Impressum und Sonstiges)
- index.htm (Startseite)
- queries.htm (Menüpunkte der verschiedenen Unterseiten)
- topogram.htm (Original-Netzplan der Bostoner Verkehrsbetriebe)

Um den einzelnen Webseiten ein einheitliches Aussehen zu geben und das Verwalten von Designdefinitionen zu erleichtern, wurde CSS eingesetzt. Diese

Definitionen finden sich in der Datei boston.css. Nachfolgend findet sich ein Ausschnitt aus der Datei:

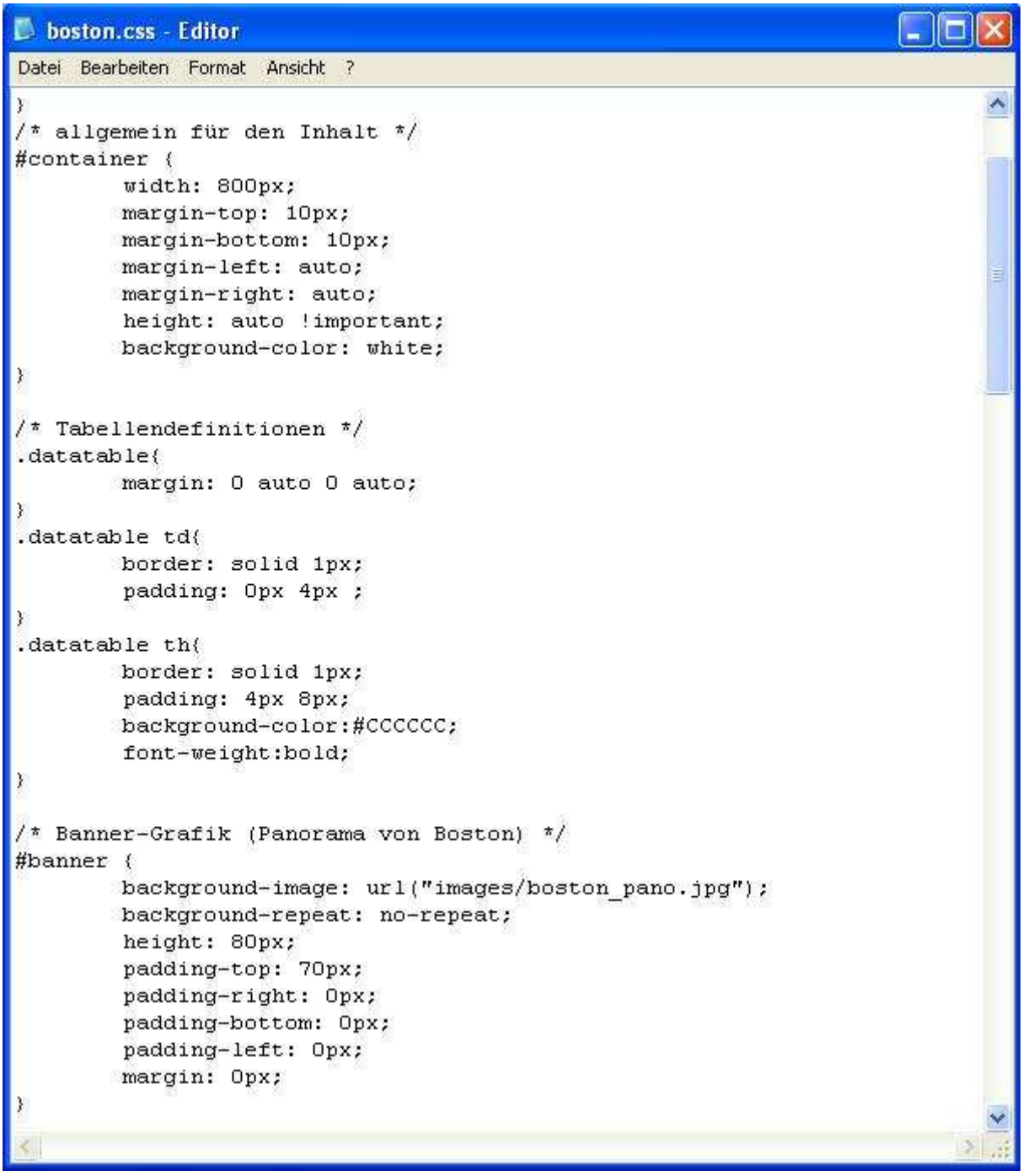

Abb. 2: Ausschnitt aus boston.css

Exemplarisch sieht man die Stildefinitionen eines Containers für allgemeinen Inhalt der Webseite, einer Tabelle und der Bannergrafik.

5.2 PHP

Alle Unterseiten der Webpräsenz, die nicht als HTM-Datei vorliegen, werden dynamisch mittels PHP erzeugt. Das heißt, dass die dargestellten Unterseiten nicht auf dem Computer des Benutzers vorliegen, sondern beim Anwählen der Seiten serverseitig erst erzeugt werden. Diese werden dann als HTML-Code dem Benutzer zurückgeschickt, damit dessen Browser den Code darstellen kann.

Im Bildbeispiel sieht man die Gegenüberstellung von einer PHP-Datei und der entsprechenden Quelltextansicht des Webbrowsers:

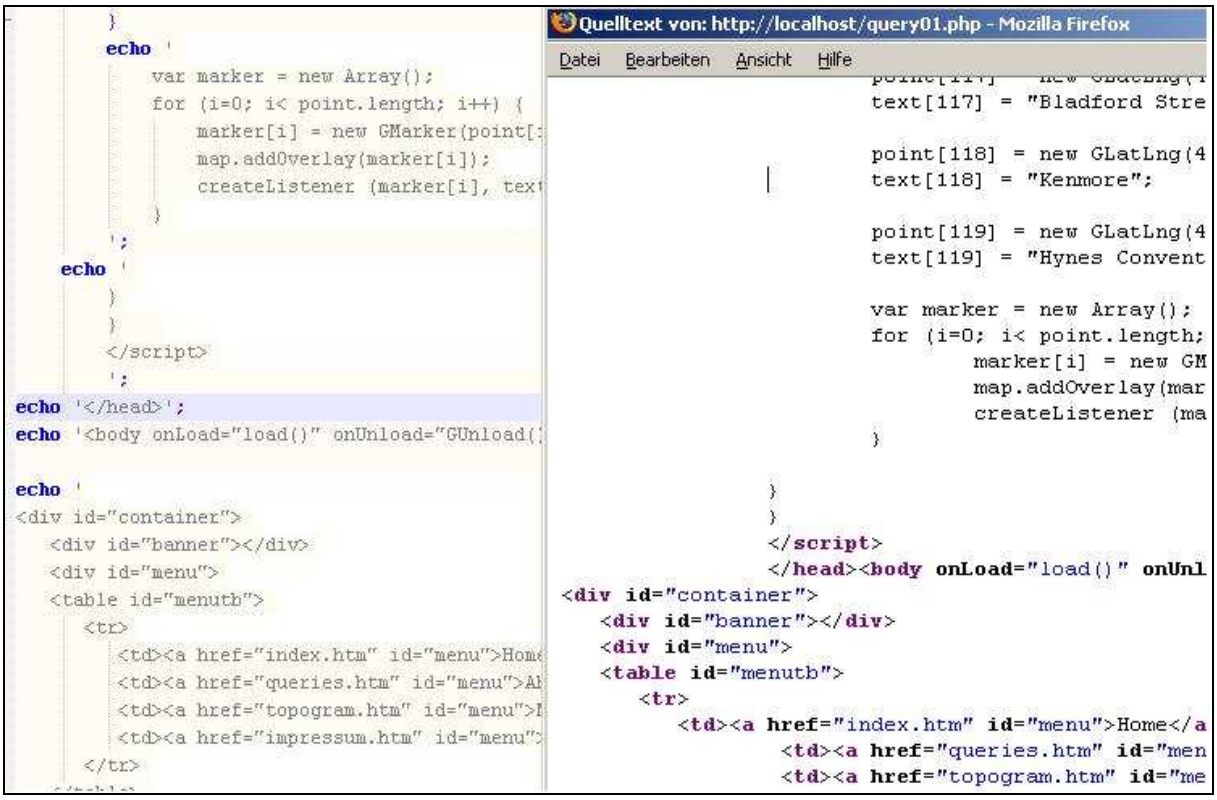

Abb. 3: Gegenüberstellung PHP und HTML 1

Links oben kann man eine for-Schleife in Javascript-Code erkennen und darunter HTML-Code. Jedoch sind beide Teile in PHP-Code (echo'...') eingebettet.

In der Quelltextansicht rechts erschließt sich, in welcher Form der Browser Code vom Webserver zurückerhält, nämlich oben ausführlicher Javascript-Code der vom </script>-Tag eingeschlossen ist und darunter den restlichen reinen HTML-Code.

5.3 SQL-Abfragen

Die SQL-Abfragen befinden sich in den jeweiligen PHP-Dateien und führen mit Hilfe spezifischer PHP-Kommandos zum Abfragen der Daten der Datenbank.

Nachfolgend werden die verschiedenen PHP-Dateien und die verschiedenen Abfragen vorgestellt.

5.3.1 query_Quincy_Alewife.php

Auf dieser Seite findet sich eine rein statische Abfrage, die zur Ausgabe einer kleinen Tabelle führt. Start- und Endpunkt innerhalb der roten Linie A sind die Haltestellen Quincy und Alewife. Ausgegeben werden die beiden Haltestellennamen, die Fahrtdauer, die Anzahl der zu passierenden Haltestellen, sowie die Reisestrecke.

Die Abfrage enthält absichtlich keine Variablen, sondern benutzt direkt die jeweiligen ids von den Haltestellen (hier 35 und 47) und die der roten Linie A (hier t_id=1) .

Die Abfrage lautet:

```
SELECT s.name,(s_distance/1000), s_time, s.id, t.name,
t.id, t.color_hex, s_order
FROM `s_t` AS st
LEFT JOIN station AS s
ON st.s_id = s.id
LEFT JOIN track AS t ON st.t_id = t.id
WHERE t_id=1 AND s_id = 35 OR s_id= 47
ORDER BY s_distance DESC;
```

5.3.2 query_Quincy_Alewife2.php

Hier wird im wesentlichen das Gleiche behandelt, wie bei der vorherigen Seite, jedoch werden die Ergebnisse auf einer Google Maps Karte visualisiert. Auf der Karte ist die komplette Linie dargestellt, die beiden Haltestellen werden zusätzlich dazu eingetragen. Zwei SQL-Abfragen liefern die dazu notwendigen Daten.

1. Abfrage (für den Streckenverlauf):

```
SELECT t_id, s_order, s.lat, s.lng, s.name, t.color_hex FROM
station AS s
INNER JOIN s_t ON s_id = id
INNER JOIN track AS t ON t.id=t_id
WHERE t_id=1
ORDER BY s_order;
```

2. Abfrage (für die Start- und Zielhaltestelle)

SELECT s.name, t.name, t.color_hex, s_order, lat, lng

```
FROM `s_t` AS st
LEFT JOIN station AS s ON st.s_id = s.id
LEFT JOIN track AS t ON st.t_id = t.id
WHERE t_id=1 AND s_id = 35 OR s_id= 47
ORDER BY s_distance DESC;
```

Ursprünglich wollte ich ab diesen beiden Abfragen auf die Verwendung von unbenannten Konstanten verzichten und damit den Unterschied zur query_Quincy_Alewife.php verdeutlichen. Jedoch wurde dann zum Beispiel der Linienverlauf nicht mehr angezeigt. Die erste Abfrage (für den Streckenverlauf) sähe folgendermaßen aus:

```
SET @line=1;
SELECT t_id, s_order, s.lat, s.lng, s.name, t.color_hex FROM
station AS s
INNER JOIN s_t ON s_id = id
INNER JOIN track AS t ON t.id=t_id
WHERE s_t.t_id LIKE @line
ORDER BY s_order;
```

Das Resultat, das die Datenbank liefert, ist mit der obigen ersten Abfrage (mit unbenannten Variablen) identisch. Ich konnte trotz intensiver Recherche nicht den Grund finden, warum aber Elemente auf der Karte dann nicht dargestellt werden. Auch weil eventuell die Funktionalität aller folgenden Abfragen gefährdet war und das Projekt kurz vor Abschluss stand, wurde dieses Problem zurückgestellt.

5.3.3 query_spreading.php

Diese Seite zeigt ganz allgemein die Verteilung der Haltestellen über das Stadtgebiet. Alleine über die farbliche Differenzierung, die der in der Datenbank gespeicherten Farbwerte der einzelnen Linien entspricht, kann der Benutzer den Linienverlauf erfassen.

Folgende SQL-Anfrage liefert die dazu notwendigen Daten:

```
SELECT s.lat, s.lng, s.name, t.color_hex
FROM station AS s
LEFT JOIN s_t AS st ON s.id = st.s_id
LEFT JOIN track AS t ON t.id = st.t_id;
```

Der Name der Haltestellen wird benötigt, um beim Klick auf die Haltestellen den Namen der Haltestelle darzustellen.

5.3.4 query_count.php

Auf dieser Seite wird die Funktionsweise des Aggregatsbefehls count() illustriert. Innerhalb der Tabelle „station“ werden die unterschiedlichen Namen aufaddiert und zurückgegeben.

```
SELECT count(name)
FROM station;
```

Hätte man die Abfrage auf eine andere Tabelle bezogen, wäre es unter Umständen nötig gewesen, doppelte Ergebnisse mit dem Befehl DISTINCT auszuschließen.

5.3.5 query_select_track.php

Der Benutzer wird aufgefordert aus einer Auswahlliste eine Linie auszuwählen, die dann auf der Karte dargestellt wird.

Die Abfrage lautet:

```
SELECT t_id, s_order, s.lat, s.lng, s.name, t.color_hex
FROM station AS s
INNER JOIN s_t ON s_id = id
INNER JOIN track AS t ON t.id=t_id
WHERE t_id=".$sel_track."
ORDER BY s_order;
```

Beim Klicken auf den Submit-Knopf wird der „option value“ des gewählten Listenelements, mit Hilfe von `$sel_track` in der Datenbankanfrage verwendet. Die „option values“ entsprichen daher den Linien-ids.

Bei der weiteren Auswahl einer Linie ist zu beachten, dass der Kartenausschnitt mitwandert. Dies wird dadurch realisiert, dass mittels des Befehls „extend“ ein Array dann mit Koordinaten gefüllt wird, wenn sich diese nicht in dem sich aufspannenden Rechteck befinden. Durch die Funktion „getBoundsZoomLevel“ lässt sich dann Google Maps anweisen automatisch die richtige Zoomstufe zu wählen, um den kompletten Inhalt des Arrays (~des Rechtecks) darzustellen. Diese beiden Befehle sind Google Maps-spezifisch.

5.3.6 query_select_length.php

Auf dieser Seite erwartet den Benutzer eine ähnliche Funktionalität wie auf der Seite zuvor, jedoch wird zusätzlich die Information auf den Bildschirm geschrieben, wie lang die Gesamtwegstrecke der auszuwählenden Linie ist. Diese wird nicht durch eine mathematische Funktion erreicht, sondern funktioniert schon deshalb, weil die Entfernung auf der jeweiligen Linie an der jeweiligen Haltestelle in der Datenbank vorliegt:

```
SELECT t_id, s_order, s.lat, s.lng, s.name, t.color_hex ,
s_distance FROM station AS s
INNER JOIN s_t ON s_id = id
INNER JOIN track AS t ON t.id=t_id
WHERE t_id=".$sel_track."
ORDER BY s_order;
```

So ist bei der Implementierung lediglich darauf zu achten, dass das zurückkommende Resultat richtig geordnet ist.

5.3.7 query_airdistance.php

In diesem Beispiel kann man sich die Entfernung (als Luftlinie) zwischen einer beliebigen Haltestelle und einem beliebigen Poi ausrechnen lassen. Der Benutzer kann mit der ersten Auswahlliste eine Haltestelle aussuchen und mit der anderen einen Point of Interest. Die beiden Listen werden hier jeweils dynamisch mit den Daten aus der Datenbank gefüllt. Das wird dadurch erreicht, dass die notwendigen HTML-Befehle für die Listenformulare innerhalb von Schleifen so oft durchlaufen werden, wie Ergebnisse vorliegen. Die HTML-Befehle die pro Listeformular nur einmal benötigt werden, befinden sich jeweils vor bzw. hinter der Schleife.

Zwei identische SQL-Abfragen liefern die Ergebnisse:

```
SELECT id, name, lat, lng FROM station ORDER BY name;
SELECT id, name, lat, lng FROM poi ORDER BY name;
```

Die Berechnung erfolgt sobald der Benutzer den Submit-Button gedrückt hat. Das Ergebnis wird in einer Textzeile ausgegeben.

5.3.8 query_nearest3.php

Auf dieser Seite bietet sich die Möglichkeit, einen Point of Interest auszuwählen und sich die 3 nächsten Haltestellen anzeigen zu lassen. Um die Auswahlliste mit Daten zu füllen, wird eine simple SQL-Anfrage ausgeführt, die den Vorangegangen entspricht und auf die nicht näher eingegangen wird.

Die für die eigentliche Funktionalität verantwortliche SQL-Query lautet folgendermaßen:

```
SELECT poi.name, station.name,
ROUND(ACOS(SIN(poi.lat*PI()/180)*SIN(station.lat*PI()/180)+
COS(poi.lat*PI()/180)*COS(station.lat*PI()/180)*COS(station.ln
g*PI()/180 -poi.lng*PI()/180))/(2*PI())*40000,3) AS 'distance'
FROM station, poi
WHERE poi.name = \"$sel_poi\"
ORDER BY distance
LIMIT 3;
```

Das Besondere an dieser Abfrage ist, dass die mathematische Berechnung der Entfernungen in der SQL-Anweisung stattfindet. Das bedeutet, dass sobald ein Point of Interest ausgewählt ist, die Entfernung zu sämtlichen Haltestellen berechnet wird. Dies geschieht innerhalb der Datenbank, der Befehl „LIMIT 3“ veranlasst aber, dass nur die ersten drei Ergebnisse an den Browser gesendet werden.
Das Ergebnis wird in einer kleinen Tabelle dargestellt.

5.3.9 query_poi_cat.php

Das Selektieren einer Kategorie der Points of Interest und die Visualisierung sind das Thema dieser Seite. Beim Aufrufen füllt eine SQL-Anfrage das Auswahlmenü mit den verschieden Kategorien der Poi's.

Die Haupt-SQL-Abfrage liefert dann den Namen, den Infotext und die Koordinaten der Suchtreffer, die dann mit eigenem Symbol der Kategorie entsprechend, dargestellt werden.

Die SQL-Abfrage lautet:

```
SELECT name, info, lat, lng, type_id  FROM poi
WHERE type_id = ".$_POST[poi].";
```

5.3.10 query_search.php

Hier kann der Benutzer innerhalb der poi-Tabelle nach dem Namen eines Point of Interest suchen. Zum Beispiel erinnert er sich nicht mehr an die ganze Bezeichnung, sondern nur noch an einen Teil davon.

Die SQL-Abfrage

```
SELECT name FROM poi WHERE (name LIKE '%$input%') ORDER BY
name ASC;
```

ermöglicht dies. Unabhängig davon, ob vor oder nach dem Eingabewert des Benutzers (der in der Variablen $input abgelegt wurde) noch Zeichen oder Leerstellen vorkommen, werden die passenden Treffer zurückgeliefert und alphabetisch angeordnet.

5.3.11 query_intersections.php

query_ intersections.php ist die erste von drei Abfrageseiten, die sich mit dem Darstellen von Umsteigemöglichkeiten befasst. In dieser ersten werden die betreffenden Haltestellen in Tabellenform angezeigt und bieten keine weitere Funktionalität.

Die Datenbankanfrage lautet:

```
SELECT s_id, s.name, t_id, count(s_id)
FROM `s_t` AS st
JOIN station AS  s ON st.s_id = s.id
GROUP BY s_id HAVING COUNT(s_id)> 1
ORDER BY s_id;
```

Dabei wird die Anzahl von Haltestellen innerhalb der Anfrage mitgezählt, aber nur diejenigen ausgegeben, die öfter als einmal vorkommen.

5.3.12 query_tracks.php

Dies ist die zweite Abfrage, die sich mit den Transfermöglichkeiten innerhalb einer Linie befasst. Zuerst wird der Benutzer aufgefordert eine der Linien auszuwählen, die im linken Teil der Seite dargestellt werden. Daraufhin wird der Linienverlauf in der Mitte aufwendig dargestellt:
Der Hintergrund hinter den Haltestellennamen ist eigentlich komplett in der Farbe der Linie, jedoch sorgt ein weißes „Hilfsbild“ dafür, dass der Hintergrund so abgedeckt wird, dass lediglich eine vertikale Linie und kleine Striche auf Höhe der Haltestellennamen sichtbar ist. Durch diesen grafischen Trick entsteht diese Darstellung.
Rechts der Linienverlaufslinie werden Links dargestellt die zeigen, an welcher Haltstelle Umsteigemöglichkeiten bestehen.
Klickt man auf diese, wird die gewählte Liniendarstellung neu abgefragt und neu dargestellt. Die dazugehörenden SQL-Queries lauten:

1) Mit dieser Abfrage werden die Daten für die Linienverlaufsdarstellung in der Mitte der Seite aus der Datenbank gewonnen:

```
SELECT s.name, s_id, t.name, t.id, t.color_hex
FROM `s_t` AS st
JOIN station AS s ON st.s_id = s.id
JOIN track AS t ON st.t_id = t.id
WHERE t_id = $sel_track
ORDER BY st.s_order ASC;
```

2) Eine zweite Abfrage ermittelt dann im nächsten Schritt wiederum die Haltestellen mit Umsteigemöglichkeiten:

```
SELECT t_id, name, color_hex
FROM s_t AS st JOIN track AS t ON st.t_id = t.id
WHERE st.s_id = '.$stationid[$r].'
AND st.t_id !='.$trackid[$r];
```

Da man die aktuell ausgewählte Linie nicht auch als auswählbaren Link dargestellt haben möchte, wird dies durch „`st.t_id !='.$trackid[$r]`“ unterdrückt. Damit werden nur die weiterführenden Linien abgefragt und dann dargestellt.

5.3.13 query_intersections2.php

Die letzte der drei Seiten, die sich mit den Umsteigemöglichkeiten befasst, ist die Seite query_intersections2.php. Sie entspricht der Abfrage von 5.3.12, jedoch wird dieses Mal keine Tabelle dargestellt, sondern Marker auf der Karte.

Die SQL-Anweisung lautet:

```
SELECT s.name, count(s_id), lat, lng, t.color_hex
FROM `s_t` AS st
LEfT JOIN station AS s ON st.s_id = s.id
LEFT JOIN track AS t ON st.t_id = t.id
GROUP BY s_id HAVING COUNT(s_id)> 1
ORDER BY s_id";
```

5.3.14 query_intersections2.php

Bei diesem Beispiel wird eine Gesamtübersicht aller Linien von Boston angezeigt. Dazu werden die einzelnen Linien, ihre Gesamtanzahl von Haltestellen, die Gesamtreisedauer und die maximale Länge tabellarisch dargestellt.

Die SQL-Abfrage sieht folgendermaßen aus:

```
SELECT t_id, t.name, COUNT(t_id), MAX(st.s_time),
MAX(s_distance/1000) AS distance
FROM `s_t` AS st
JOIN track AS t ON st.t_id = t.id
GROUP BY name
ORDER BY t.name;
```

Neben `COUNT, wird` nun auch `MAX` als Aggregatfunktion verwendet. Sie liefert den größten Wert eines Datenfeldes einer Abfrage. Die beiden Aggregatfunktionen beziehen sich dabei auf die `GROUP BY` – Anweisung.

5.3.15 query_distance.php

Diese Seite bietet dem Benutzer die Möglichkeit sich zu informieren, wie lange die Fahrt zwischen zwei Stationen einer Linie dauert.
Zwei Abfragen liefern dazu die Funktionalität, wobei die Erste wiederum nur für das Füllen des Auswahlmenüs zuständig ist und nicht näher beschrieben werden soll, da sie in vorangegangenen Abfragen in dieser Form bereits vorkam.

Die zweite Abfrage lautet:

```
SELECT s.name, s_distance, s_time, s.id, t.name, t.id,
t.color_hex, s_order
```

```
FROM `s_t` AS st
JOIN station AS s ON st.s_id = s.id
JOIN track AS t ON st.t_id = t.id
WHERE t_id = $sel_track
AND s_id = $startpoint or s_id= $endpoint
ORDER BY s_distance DESC;
```

In Abhängigkeit der beiden gewählten Haltestellen wird die notwendige Reisedauer abgefragt. Die Berechnung der tatsächlichen Dauer wird außerhalb des SQL-Anfrage berechnet. Notwendig ist eine Fallunterscheidung, damit keine negativen Werte entstehen können. Ausgegeben werden die Dauer der Fahrt, sowie die Anzahl an den Haltestationen, an denen man vorbeikommt (einschließlich der Zielhaltestelle).

5.3.16 query_sql.php

Diese Seite bietet dem Anwender die Möglichkeit, eigene Datenbankabfragen zu formulieren. Dazu kann er sein SQL-Query in ein Textfeld schreiben und mit einem Submit-Button abschicken.

Dies sind die beiden Code-Zeilen, die für diese Funktionalität verantwortlich sind:

```
$query = $_POST[user_query];
$result = mysql_query($query);
```

Fehler werden lediglich dadurch abgefangen, dass der Anwender die Meldung bekommt, dass die Datenbankabfrage keine Ergebnisse geliefert hat. Er sollte also die SQL-Befehle, bzw. deren Schreibweise kennen.

5.3.17 query_tables.php

Mit Hilfe der Seite query_tables können alle fünf in der Datenbank existierenden Tabellen dargestellt werden. Per Radio-Button wird die gewünschte Tabelle ausgegeben.

Die SQL-Abfrage lautet:

SELECT * from $sel_table;

Mittels einer Vorschleife und dem Befehl „mysql_field_name“ werden die Tabellenüberschriften mit den richtigen Spaltennamen aus der Datenbank abgefragt und eingetragen.

5.3.18 query_init.php

Die letzte Seite meines Projektes sollte dem Benutzer die Möglichkeit bieten, mit Hilfe der boston.sql Ladedatei die Datenbank neu aufzuspielen, falls sie fehlerhaft sein sollte. Jedoch ist es nicht ganz einfach die notwendigen Rechte zu gewähren, insofern wurde diese Funktionalität nicht vollständig implementiert.
Auf der Seite erscheint die Frage, ob der Benutzer sich sicher ist, diesen Schritt ausführen zu wollen. Als Auswahl werden jeweils ein mit „Ja“ und „Nein“ beschrifteter Button angeboten, jedoch kommt es beim Drücken von Ja sofort zu einem Fehler. Beim Klicken auf Nein, wird man auf die queries.htm-Übersichtsseite zurückgeleitet.

5.4 Validierung der Seiten

Alle Seiten (PHP, HTML und CSS) wurden auf ihre W3C-Konformität hin überprüft und wurden als fehlerfrei eingestuft.

6 Fazit

6.1 Ergebniskritik

Im Einzelnen zeigen die Abfrageseiten dem Benutzer anschaulich die verschiedenen Darstellungsmöglichkeiten, die sich beim Verwenden der benutzten Software- oder Codekomponenten bieten. Jedoch ist ein Mangel an Komplexität ersichtlich, der zum Teil auf den wenigen Tabellen in der Datenbank zurückzuführen ist. Es war nicht einfach, grundlegend unterschiedliche SQL-Abfragen zu formulieren, weshalb zum Teil versucht wurde, auf unterschiedliche Arten ähnliche Sachverhalte darzustellen. Ein Ergänzen der Datenbank wäre deshalb notwendig und im Nachhinein wünschenswert gewesen.
Auch wären einige Abfragen innerhalb bestimmter Seite zusammenfassbar gewesen, jedoch blieb gegen Ende des Projektes keine Zeit mehr, Abfragen komplett neu zu erstellen. Diese Problematik wurde durch das gleichzeitige Arbeiten an den verschiedenen Komponenten, wie u.a. CSS, PHP und Javascript und auch der Dokumentation verursacht. Ein besseres Zeit- und Projektmanagement hätten dem gut entgegenwirken können.

6.2 Zeitaufwand

Konzeption:	4 Stunden
Seitendesign:	4 Stunden
Projektanalyse:	2 Stunden
Einarbeiten in php / css:	12 Stunden
Umsetzung:	50 Stunden
Fehlerbeseitigung:	25 Stunden
Dokumentation:	18 Stunden

7 Quellen

7.1 Literatur

- http://de.selfhtml.org
- SQL Grundlagen und Datenbankdesign (Universität Hannover)
- www.php.net

7.2 Grafiken

Die Bannergrafik (Panoramafoto von Boston) wurde dem zentralen Medienarchiv Wikimedia Commons entnommen und unter GNU-Lizenz für freie Dokumentation veröffentlicht.

Die verwendeten Punktsignaturen für die Metro und die Poi wurden von einem befreundeten Grafiker entworfen und mir wurde das Recht eingeräumt, sie für dieses Projekt zu benutzen.