**Web Application Development**

**Report**

**Project Title*:*** *‘BABURE: Railway system’*

**Report**

**Submitted to:** Hannu Kähkölä/ Maisa Mielikäinen

By:

Dawit Afework Yohannes

0800385

January, 2009

**Introduction**

BABURE is sample PHP/MySql web application that allows travellers to view train schedules. It bases its scheduling on an artificial railway system in Ethiopia but can be changed for other areas as well. The data it uses are hypothetical as it mainly focuses on demonstrating usage of PHP and MySql in maintaining train and schedule information.

This paper presents an introduction about BABURE and documents the database schema and decisions taken to construct the database in detail.

**Functionality**

BABURE has four modules: namely, traveller, administration, salesperson and manager. There are four categories of user types each pertaining to the modules. Traveller users use the traveller (home page) module to query for schedules. No login is required for traveller users. Administrators, ticket salespersons and managers use the URL, <http://domainname/admin> to access their module (for example, [www.babure.com/admin](http://www.babure.com/admin)) and should provide proper user details to log in.

However, not all of these modules are complete. For demonstration purpose, only some menu Items in each module are working.



Fig.1. BABURE schedule search

The interaction between the system and users is summarized by t he use case diagram on Fig 2. In this project, the focus has been on traveller and Admin users so most of the functionalities for these users have been implemented in BABURE.

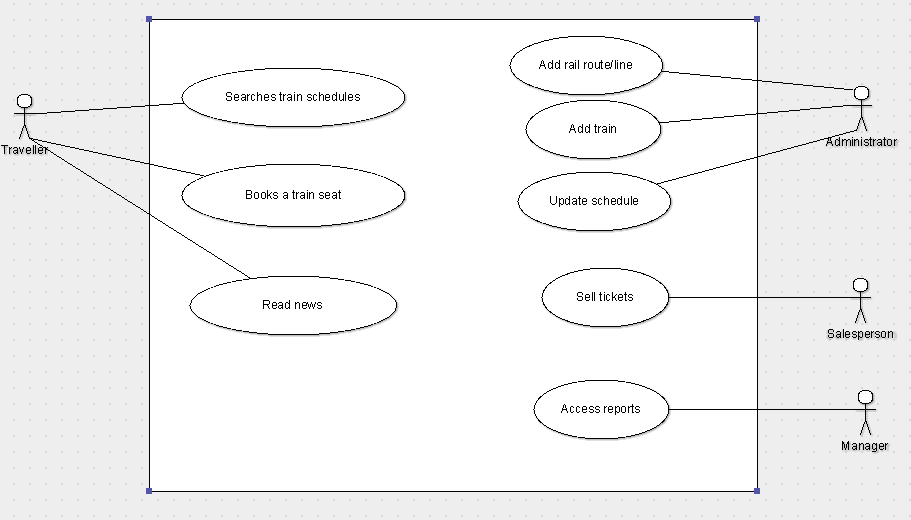
****

Fig.2. Simple Use case diagram for Babure

**PHP code samples**

The following are code snippets from BABURE.

* Checking if a submit button named btnsearch has been clicked.

***if(isset($\_GET['btnsearch']))***

***{***

***include("quick\_search\_inc.php");***

***}***

* Connecting to MySql database

***<?php***

***$hostname\_dbconnection = "localhost";***

***$database\_dbconnection = "babure";***

***$username\_dbconnection = "root";***

***$password\_dbconnection = "";***

***$dbconnection = mysql\_pconnect($hostname\_dbconnection, $username\_dbconnection, $password\_dbconnection) or trigger\_error(mysql\_error(),E\_USER\_ERROR);***

***?>***

* Checking if the login account exists, and if so, picking the right user type and the right page for the user type. Session is started for users that are logged in and a session variables that hold username, password and type are created.

***<?php***

***session\_start();***

***include("../Connections/dbconnection.php");***

***$userid=$\_POST['userid'];***

***$password=$\_POST['password'];***

***$getuser="SELECT \* from administrators WHERE (userid='$userid' And password='$password')";***

***// the type of administrator has not been checked***

***$result= mysql\_query($getuser);***

***if (mysql\_num\_rows($result) >= 1 )***

***{***

***$row=mysql\_fetch\_array($result);***

***$usertype=$row['type'];***

***$properuser=$row['userid'];***

***$\_SESSION['properuser']=$properuser;***

***$\_SESSION['account\_type']=$usertype;***

***$\_SESSION['password']=$password;***

***// Pickin the right kind of user and page for each admin type....***

***switch($usertype)***

***{***

***case 'admin':***

***header("Location: administration.php");***

***break;***

***case 'sales\_person':***

***header("Location: sales\_person.php");***

***break;***

***case 'manager':***

***header("Location: manager.php");***

***break;***

***}***

***exit;***

***}***

***else***

***{***

***header ("Location: index.php?ermsg=Sorry, wrong username and /or password");***

***}***

***?>***

**Database Schema**

For websites like Babure, the backend is the backbone of the whole application. It is necessary to plan and think about the data entities required and their relationships so that data will be captured in a way that is suitable for querying and easier computation.

I have spent a chunk of my time on this project on trying to come up with a solid conceptual design of a railway system that can give proper results for any possible SQL query regarding schedules.

The major Entities of BABURE with their attributes are:

* **RailLines (rail routes):** hold information about rail routes in the rail system (Babure).
  + Raillines(id,name,origin\_id,destination\_id)
* **Railstations :** This holds station information throughtout Babure
  + Railstations(id,name)
* **Trains :** Holds information about trains
  + **Trains**(id)
* **Administrators:** Holds information about administrators, salesperson and managers
  + **Administrators**(userid,password,type)

There are other entities that are the result of **Many to Many** relationship between the major entities. They are all depicted on the fig 3. ER diagram.

The diagram depicts two distinct Many to Many relationships**.**

* *Binary relationship* between **Raillines** and **Railstations.**  Rail lines have many rail stations on them, and a station can be on many rail lines. This results on a many to many relationship named **StationLines**
* *Tertiary relationshiop* among **Raillines**,**railstations** and **trains.**  This relationship is given the name Departuretimes as it holds fixed departure/arrival time information for each **rail station** on a certain train’s journey on a **railline**.

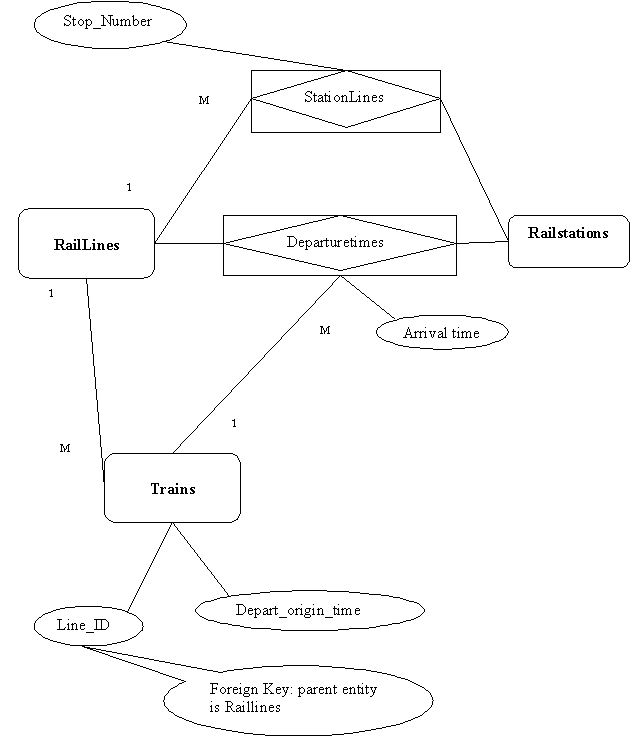


Fig 3. ER diagram

**Sample SQL queries**

* Getting the names of all rail stations in alphabetical order.

SELECT \*

FROM railstations

ORDER BY name ASC

* Counting trains and outputting the result plus one as nxt. This will be used as ID for next train entry ( the Ids of trains does not auto increment)

SELECT count(\*)+1 as nxt

FROM trains

* Calculating the time difference (duration) between arrival and departure between departure and arrival stations. Both stations should be on the same route/line (the right\_line), one has to be the departure station(station from) and the other has to be the arrival station (station to), the time difference (duration) is also dependent not only on the route but also on the kind of train, thus the need to check of the train as well.

SELECT TIMEDIFF(E.arrival\_time,S.departure\_time) as duration,E.departure\_time

FROM departuretimes S,departuretimes E

WHERE (S.line\_id='$right\_line') AND (E.line\_id = '$right\_line' )AND (S.station\_id='$station\_from' )AND (E.station\_id='$station\_to' )AND (S.train\_id='$train' AND (E.train\_id='$train')

* Getting the train id, departure station name and departure time for a query about a schedule from a certain station to another. The SQL query combines two tables: departuretimes and railstations, using the equality of station ids in the railstation table and station\_id (foreign key) in the departuretimes table.

SELECT train\_id,name,departure\_time

FROM departuretimes,railstations

WHERE departuretimes.line\_id='$right\_line' AND railstations.id=departuretimes.station\_id AND departuretimes.station\_id='$station\_from'

* Returning a record from administrators table where userid and password are equal to the variables indicated (used to authenticate log in details from users).

SELECT \*

FROM administrators

WHERE (userid='$userid' And password='$password')

**Conclusion**

This project has been quiet interesting in the sense that it poses a very data intensive and complex problem. The core of the problem is on designing a concrete database schema that enables the application to have proper answers for every query imaginable. In this regard, BABURE has a good database design that can support such a train schedule system well.

It is my hope that BABURE has demonstrated the PHP web development and database design skills that I need to possess as an IT professional.

**Usage Note**

* Unzip Babure
* Copy and paste BABURE folder in your web servers document root directory (e.g in the www directory)
* Create the Babure database by using the SQL statements in the babure\_database.txt file found in the BABURE folder (copy all the content of the file and import it into mysql).
* If you need to change username and password settings for connection to the database, make the modifications on **dbconnection.php** file found in connections sub directory of BABURE folder