Lab II Section 3 — intro descr.

3.1.1 Database (Odean Maye)

— see fig 1

The database must contain tables that store values that correspond to each particular component of the system.

1. Time must be stored in the database in the format HHMMSS.
2. Date must be stored in the database in the format as YYYYMMDD.
3. The database must contain a table named trains that stores the following information in fields :
   i. The unique identifying number stored as a SMALLINT.
   ii. The active status stored as a TINYINT.
   iii. A reference to the schedule id.
4. The database must contain a table named users that stores the following information in fields:
   i. The unique identifying number stored as a MEDIUMINT.
   ii. The given name as a TINYTEXT.
   iii. The username stored as a TINYTEXT.
   iv. The password stored as a TINYTEXT.
   v. A reference to an admin level defined in 3.1.1.10.i .
5. The database must contain a table named stops with required fields as defined by the Google Transit Feed Specifications at https://developers.google.com/transit/gtfs/reference#stops_fields
   i. The unique identifying number stored as a SMALLINT.
   ii. The name stored as a TINYTEXT.
   iii. The latitude stored as a FLOAT.
   iv. The longitude stored as a FLOAT.
6. The database must contain a table named adverts that stores the following information in fields:
   i. The unique identifying number stored as a SMALLINT.
   ii. A reference to a stop id defined in 3.1.1.5.i .
   iii. The advert name stored as a TINYTEXT.
   iv. A reference to a user id defined in 3.1.1.4.i .
   v. The category stored as a TINYTEXT.
   vi. The description stored as a MEDIUMTEXT.
   vii. The start date stored as a DATE.
   viii. The end date stored as a DATE.
   ix. The start time stored as a TIME.
   x. The end time stored as a TIME.
   xi. The image url stored as a TEXT.
   xii. The TIMESTAMP.
7. The database must contain a table named occupancy that stores the following information in fields.
   i. The unique identifying number stored as a BIGINT.
   ii. A reference to a stop id defined in 3.1.1.5.i .
   iii. The number of embarks stored as a SMALLINT.

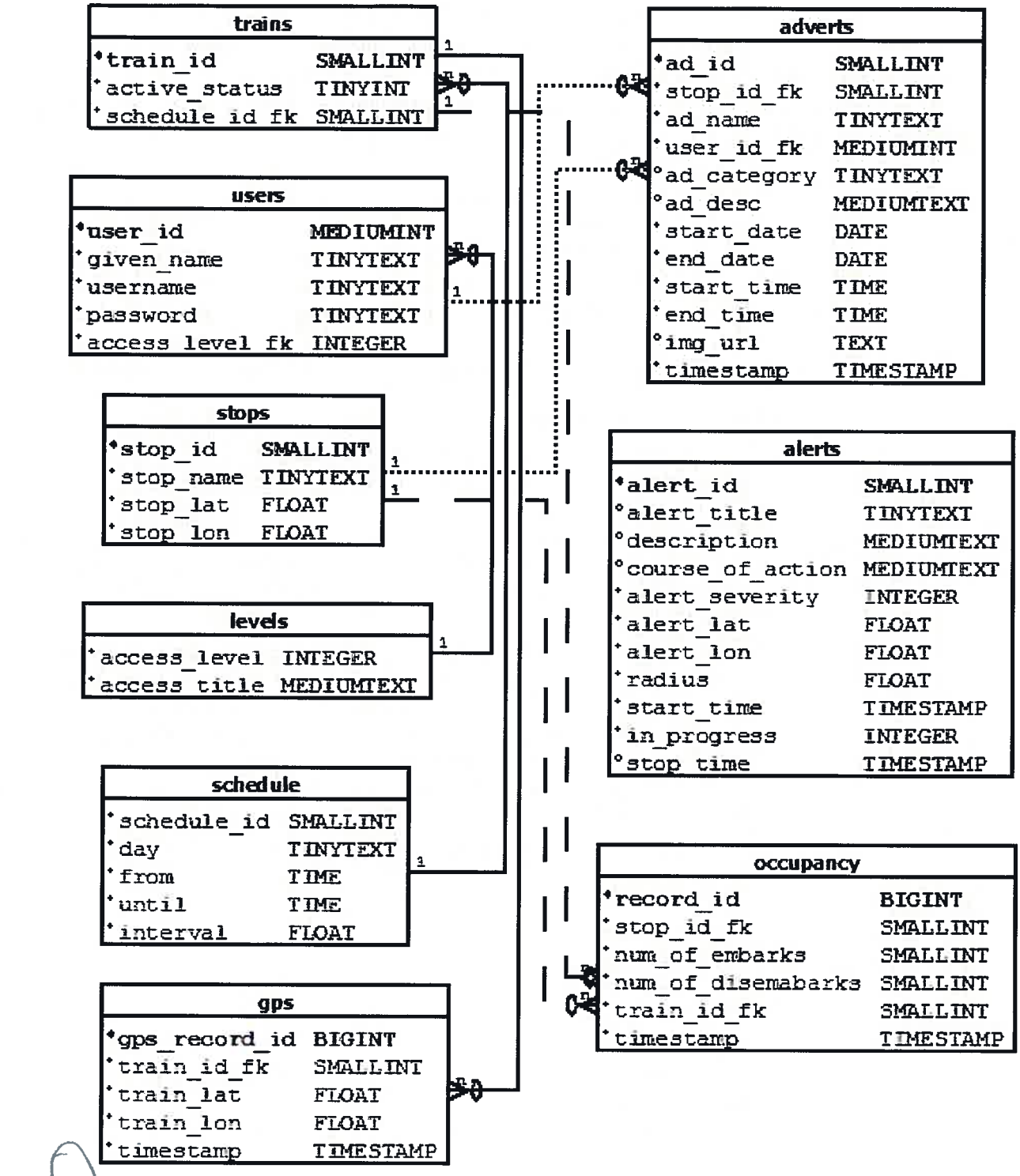


Figure . Database Schema

vii. Provide output to Ridership Trend Report function in the form of non-negative integers. (3.1.4.6)

2. **Delay Impact Calculator**

Provide an interface for the Web Application Engine to request a Delay Impact report. Provide the ability to query the Current ITS database for the most recent simulated GPS location value of active trains.

i. Provide the ability to identify a GPS coordinate to include:
   - Precede South latitudes and West longitudes with a minus sign.
   - Latitudes range from -90 to 90.
   - Longitudes range from -180 to 180.

ii. Provide the ability to identify a date as specified in 3.1.2.1.i.a

iii. Provide the ability to query the "trains" table for active trains during the date specified in 3.1.2.2.ii.

iv. Provide the ability to query the Current ITS database "GPS" table for past simulated arrival times at the station during the date. (3.1.1.11)

v. Provide the ability to query the "schedule" table. (3.1.1.9)

vi. Provide the ability to compare the values specified in 3.1.2.2.iv and 3.1.2.2.v and store the average variance.

vii. Provide the ability to query the "alerts" table for any active alerts and their severity level. (3.1.1.8)
   a. Specified must be used in its calculation of delay using the following categories and associated Delay intervals.
      - 1 – Local Warning – No Delay
      - 2 – Local Problem – 10 Min Delay
      - 3 – System Warning – 20 Min Delay
      - 4 – System Problem – 30 Min Delay
      - 5 – System Failure – 50 Min Delay
      - 6 – System Shutdown - Interpreted as full stop of trains.

viii. Provide the ability to apply the delay interval to the average variance stored in 3.1.2.2.vi

ix. Provide the ability to compare the expected value of time-to-arrival between the calculated variance and current GPS position to the HRT schedule.

x. Provide output to Train Data Report module in the form of a time value. (3.1.4.7)

3. **Ontime Performance Reporting**

Provide an interface for the Web Application Engine to request a Delay Impact report.

i. Provide the ability to identify a date range as specified in 3.1.2.1.i

ii. Provide the ability to identify a stop ID as specified in 3.1.2.1.ii

iii. Specified must be used to query the Current ITS database "GPS" and "STOPS" table for past simulated arrival times at the station. (3.1.1.11) (3.1.1.5)

iv. Provide the ability to query the "schedule" table. (3.1.1.9)

v. Provide the ability to compare the values specified in 3.1.2.3.iii and 3.1.2.3.iv and return the variance.

vi. Provide output to Train Data Report module the average variance, in the form of a time value. (3.1.4.7)

3.1.3 Test Harness (Akeem Edwards)

The Test Harness will be used to demonstrate the current ITS prototype. This will be a standalone application that will maintain communication with the web application engine. The following functional requirements must be met:

1. **GPS Data Control Module**

Current ITS prototype will use a software module that will take a range of GPS coordinates then creates a virtual route. The GPS Control module then retrieves virtual stops from the database and determines which virtual stops correlate each route created. The GPS control Module will also assign GPS coordinates to each active train along the virtual route. The following functional requirements must be met:

i. Provide the ability to access the virtual stops in the database to determine what GPS coordinates to which stops are assigned (3.1.1.4).

ii. Provide support GPS coordinate parameters:

iii. Input parameters for single GPS coordinate are floating point values for latitude and longitude (3.1.2.2.1.1.1).

iv. Input parameter for GPS coordinate set in an array structure to represent a virtual route.

v. Provide the ability to assign a GPS coordinate to each virtual train active:

   a. Each coordinate must translate to the correct virtual route.
   b. Each coordinate must be updated in half a minute intervals.

2. **Ridership Data Control Module**

In the Current ITS prototype the Ridership Data Control Module will generate virtual riders to for the prototype demonstration. The Ridership Data Control will assign virtual ridership to each active train. This Module will generate virtual rider numbers at each stop representing departures and arrivals, and compare these numbers to assign the current amount of riders on each train. The following function requirements must be met:

i. Provide the ability to generate virtual riders at each stop.
ii. Must utilize mathematical probability distributions based on past ridership data to define:
   a. The amount of arrivals on each train.
   b. The amount of departures on each train
   c. Must utilize probability distributions to estimate amount of virtual riders generated on days with events.

3.1.4 Web Application Engine

1. **Alert Module (Chris Coykendall)**
   The Alert Module will provide the ability view and/or manage alerts on the Current ITS website user interfaces.

   i. Provide the ability to SQL query the Alerts database table to obtain a list of ongoing outages and exceptions to normal vehicle operations and display them (during the initial page load, select only ongoing alert records which have the ongoing flag set in the table.) (Requirement 3.1.4.11)

   ii. Provide the ability to SQL query the Alerts database table to create or modify alerts for outages and exceptions to normal vehicle operations. (Requirement 3.1.1.4, 3.1.4.11)

   iii. Provide the ability for a user viewing HRT GUI to complete the following fields:
      a. Title
      b. Begin
      c. End (Estimated if future)
      d. Description
      e. Course of Action
      f. Submit (Triggers Requirement 3.1.4.1.4)

   iv. Provide ability to generate an Alert via SQL query to the Alerts database table with an initialized end time stamp of null and ongoing the ongoing flag set to true. (Requirement 3.1.4.11)

   v. Provide an additional Close button next to events that are ongoing in the HRT GUI view, which will set the end timestamp for the particular Alert to the current date/time via SQL query in the Alerts database table. (Requirement 3.1.4.11)

   vi. Provide the ability to view any fetched Alerts in any web interface (HRT, Business or Rider.)

2. **Feedback Module (Chris Coykendall)**
   The Feedback Module will provide a mechanism to accept user feedback regarding Current ITS from the website user interface, and email the obtained feedback to the development team.

   i. Provide ability for a user to complete the following fields:
      a. Name (REQUIRED)
      b. E-mail
      c. Subject (REQUIRED)
      d. Message
      e. Submit Button (Triggers Requirement 3.1.4.2.2)
      f. Reset Button (Triggers Requirement 3.1.4.2.4)

iv. Provide the ability to SQL query the Adverts database table to create the new Event. (Requirement 3.1.4.11)
v. Provide the ability to reset the fields of the form by clearing any previously entered input. (Requirement 3.1.4.5.2)
vi. Provide an additional Close button next to Events that are ongoing in the HRT and Business GUI views, which will set the end time stamp for the particular Event to the current date/time via SQL query in the Adverts database table. (Requirement 3.1.4.5.1, 3.1.4.11)

6. **Ridership Trend Report (Brian Dunn)**
The Ridership Trend Report will provide ridership information for display on all three GUI Frameworks (Requirement 3.1.1.4).

   i. Provide customization of data for the different authentication levels through the GUI Framework. (Requirement 3.1.1.4)
   ii. Provide ability to select a date (YYYY -MM-DD) for which to view hourly report.
   iii. Provide the ability to select a start date (YYYY-MM-DD) and end date (MM-DD-YYYY) display summarized report averaging data over the timespan.
   iv. Provide a table output with ridership information: time, number of departures, and number of arrivals.
   v. Must interface with DE to retrieve real-time data for reports. (Requirement 3.1.4.11.2)

7. **Train Data Report (Brian Dunn)**
The Train Data Report provide on-timer performance data for display on the HRT GUI (Requirement 3.1.1.4.2). The following functions shall be provided:

   i. Provide ability to select a date (YYYY-MM-DD) for which to view hourly report.
   ii. Provide the ability to select a start date (YYYY-MM-DD) and end date (YYYY-MM-DD) display summarized report averaging data over the timespan.
   iii. Provide a table output with train performance information: time, on-time percentage, and delay time.
   iv. Must interface with DE to retrieve real-time data for reports. (Requirement 3.1.4.11.2)

8. **Business Ad Campaign Module (Brian Dunn)**
The Business Ad Campaign Module will allow the Businesses GUI and HRT GUI to create and modify advertisement campaigns.

iii. Incorporate a method for a user to retrieve their username.
iv. The capability to allow user to reset the password.
v. An additional user information update screen allowing a user access to change personal information is also required
vi. Contain a web page to allow administrators to perform administrative tasks to the user accounts to include the following
    a. Edit user information
    b. Change applicable access group
    c. Reset a user's password
vii. Administrative capability to create user groups and manage members.

10. **Authentication (CJ Deaver)**

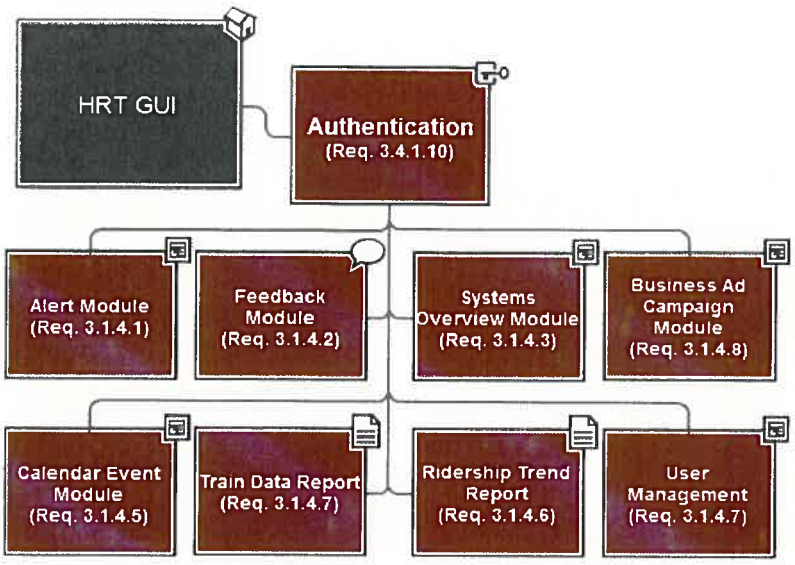
Provide a methodology for security control utilized throughout the application. The following are the minimum requirements:

i. A one-factor authentication mechanism for securely authorizing application access.
ii. Token generation or other means of identifying users.
iii. Access control mechanism controlling account access within the application.
iv. Security time-out feature to log users out after a set amount of time
v. Include an account locking mechanism in the event of multiple failed login attempts.
vi. Logging capability recording the following:
    a. Login time
    b. Login location
    c. Authentication success or failure
    d. Page requested

11. **Data Integration Utility (CJ Deaver)**

Provide an interface for connecting the web application to the different data sources to pass data throughout the system with the following requirements:

i. DB Interface
    a. Requires the capability to open and close data stream connections
    b. Must complete transfer queries and results between the database and the Web Application Engine for the following modules:
        1. Alert Module (3.1.4.1)
        2. Feedback Module (3.1.4.2)
        3. System Overview Module (3.1.4.3)
        4. Calendar Event Module (3.1.4.5)
        5. Ridership Trend Report (3.1.4.6)



iii. Business Graphical User Interface

The Business GUI will provide the following modules to users logged in with Business Owner permissions: