EECS 647 Project Report 2

Plan for Implementation

Autocross Data System

Team Members

Chris Pham, Sean Boeck, Cesar Dos Santos

**Development Tools**

Code for the program will be developed using Eclipse with the PHP plugin. To run and test the code, WampServer will be used to locally host the website and database.

**Web Pages**

The Autocross Data System project will include a total of four web pages. The first web page shall be a home page, which will include website navigation, a short summary about the website, and a user login form. The second web page will be used to query the database to find certain information. This information is public, thus accessing this web page does not require user login. The user will retrieve information from the database by filling out a form and submitting it. That form will be processed, a database query will be made, and the result will be returned and displayed to the user. The third webpage will be a hall of fame page. Its purpose is to show every winner of each class of each region for every year. The last web page will be for inserting and updating data in the database. This page must require user login because we must restrict who can modify the database. Registered users will modify the database by entering data into a form and submitting it. That data will be processed into a query to insert or update the database, and the user will be notified if the update was successful.

Every web page shall include the user login module, which tells the user whether or not they are logged in. If they are not logged in, then a user login form is presented.

**Database Queries**

In order to simplify some of the database queries, a view named SUM\_POINTS is created. This view is a relation of drivers and their total points in a class per region per year. SUM\_POINTS is created using the following query:

CREATE VIEW SUM\_POINTS AS

SELECT Name, SUM(Points) Total\_points, Class\_name, Region\_Name, Year

FROM CLASS, HOST, EVENT, DRIVERS

WHERE CLASS.Event\_ID = HOST.Event\_ID

AND HOST.EVENT\_ID = EVENT.Event\_ID

AND CLASS.Member\_ID = DRIVERS.Member\_ID

GROUP BY DRIVERS.Member\_ID, Region\_name, Year

Listed below is the information that users will be able to retrieve through the query web page and the query that gives that information.

* Find all events hosted by a given *region* and *year*.

SELECT EVENT.Event\_ID, EVENT.Location, EVENT.Date

FROM EVENT, HOST

WHERE EVENT.Event\_ID = HOST.Event\_ID

AND Region\_name = '*region*'

AND Year = '*year*'

ORDER BY EVENT.Event\_ID

* Find all drivers who attended a given *Event\_ID*.

SELECT Name, Time, Points, Class\_name, Car\_Model, Car\_Num

FROM CLASS, DRIVERS

WHERE CLASS.Member\_ID = DRIVERS.Member\_ID

AND Event\_ID = '*Event\_ID*'

ORDER BY Class\_name, Time

* Find all events that a driver with the given *name* has attended in the given *region*.

SELECT EVENT.Event\_ID, Location, Date, Time, Points, Class\_name, Car\_model, Car\_num

FROM EVENT, CLASS, HOST, DRIVERS

WHERE EVENT.Event\_ID = CLASS.Event\_ID

AND EVENT.Event\_ID = HOST.Event\_ID

AND CLASS.Member\_ID = DRIVERS.Member\_ID

AND Name = '*name*'

AND Region\_name = '*region*'

ORDER BY EVENT.Event\_id

* Find the current standings (every driver’s current total points this year) of a given *region*.

SELECT Name, Total\_points, Class\_name

FROM sum\_points

WHERE Year = DATE\_FORMAT(CURDATE(), '%Y')

AND Region\_name = '*region*'

ORDER BY Class\_name, Total\_points DESC

For the hall of fame web page, the following query is used to find the history of winners. The results for the current year aren’t included, since the year may not be finished.

SELECT \*

FROM SUM\_POINTS s

WHERE NOT EXISTS

(SELECT s.Total\_points, s.Class\_name, s.Region\_name

FROM SUM\_POINTS

WHERE Total\_points > s.Total\_points

AND Class\_name = s.Class\_name

AND Region\_name = s.Region\_name

AND Year = s.Year)

AND NOT Year = DATE\_FORMAT(CURDATE(), '%Y')

ORDER BY Year, Region\_name

Listed below are queries that will be made to update the database.

* Insert a new member into the DRIVER relation given the *Member\_ID* and *Name*.

INSERT INTO DRIVERS VALUES ('*Member\_ID*', '*Name*')

* Insert a new event into the EVENT relation and insert what region hosted that event in the HOST relation given the *Event\_ID*, *Region\_name*, *Year*, *Location*, and *Date*.

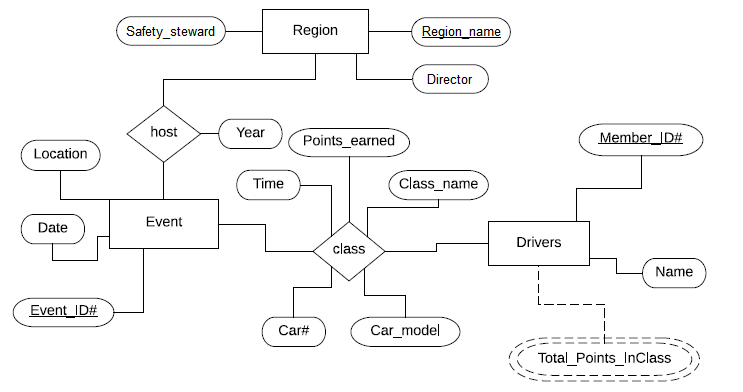
INSERT INTO EVENT VALUES ('*Event\_ID'*, '*Location*', '*Date*');

INSERT INTO HOST VALUES ('*Event\_ID*', '*Region\_name*', '*Year*')

* Insert a driver’s results from an event into the CLASS relation given the *Member\_ID*, *Event\_ID*, *Time*, *Points*, *Class\_Name*, *Car\_Num*, and *Car\_Model*.

INSERT INTO CLASS VALUES ('*Event\_ID*', '*Member\_ID*', '*Points*', '*Time*', '*Class\_name*', '*Car\_num*', '*Car\_model*')

**Changes to the ER-Diagram**

The ER-Diagram has been revised to avoid a bad entity. The attributes Director and Safety\_steward have been added to Region.