

CPSC 304 Project Cover Page

Milestone #: 1

Date: 10/06/2023

Group Number: 38

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Michael Tham	36082469	s6f2k	mtham01@student.ubc.ca
Christopher Wong	78318987	k4g2g	chw2305@student.ubc.ca
Maureen Marthias	83217794	w5t9j	mmarthia@student.ubc.ca

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia.

2. A brief project description answering these questions:

- a. What is the domain of the application? Describe it. The domain of an application refers to the area of knowledge your application resides in. For example, if I am making an application for a hospital, the domain would be something like healthcare/patient management/logistics (it would depend on what the application is trying to do).

We are lovers of fast cars and competition so supporting and watching Formula 1 (F1) is one of our favourite pastimes. Being avid F1 fans, we enjoy looking at the standings of current and past races, seeing how our favourite drivers do. However, we find the official F1 website very intimidating and cluttered with a plethora of advertisements, news feeds, and different tabs. We are simple people that prefer a minimalist approach.

With that being said, we would like to create an application that displays relevant F1 race data and standings. A user would be able to view each season described by the year the season was held (e.g. '2017') and view a multitude of data - races (location, drivers for that race, driver standings of that race), the drivers for that season and teams for that season just to name a few.

- b. What aspects of the domain are modelled by the database? In answering this question, you will want to talk about what your project is trying to address and how it fits within the domain. It is likely that in the process of answering these questions you will bring up examples of a real-life situation that the application could be applied to.

We are trying to model F1 race weekends and how each weekend affects the standings of the current season. For example, when the first weekend of the season starts, users can see all the necessary information before and after the race. Before the race, users can see the current standings, the track being used, the racing drivers, etc. After the race, a user can see all races' results (i.e., Grand Prix, Qualifying, Sprint, Practice) and how the new standings get updated after each race.

Our project has **13** entities:

Seasons - represents seasons in F1 (e.g. 2021, 2022 etc.)

GrandPrix - represents each Grand Prix race with three attributes (id, starting grid, and datetime). Every time a grand prix is finished, it will produce a result that will contribute to the Standings entity.

Results - is a weak entity that is dependent on the GrandPrix entity. This entity represents each GrandPrix race's results, where we have attributes for driver position and driver name for every GrandPrix race.

Standings - represents the standings, where each standing after each race is different.

Races - represents each race, where we have raceID as the primary key. Race have ISA entities which are:

GrandPrix - a race type that will affect seasonal standings

Qualifying - a race type that will affect GrandPrix starting grid

Sprint - a race type that allows drivers to gain extra points based on their finish position, potentially allowing for them to gain an extra edge in the seasonal standings

Practice - a race type that won't affect any other entities as it will only be treated as a practice race by the drivers but driver's timings are still recorded.

Tracks - represents each race location. (e.g. Singapore, Monza etc.)

Constructors¹ - represents the constructors in the season. A constructor has two drivers, some engineers, and sponsors. A constructor can have no partner.

Drivers - represents the drivers in the season.

Partners - represents the partners for constructors.

3. Database specifications: (3-5 sentences)

- a. What functionality will the database provide? I.e., what kinds of things will people using the database be able to do.

Users will be able to use the database to update information on current F1 races to let them keep track of results and standings from the current F1 season as well as previous ones in a nice and organised fashion. For example, they could see what constructors racers used to drive for before their current ones, or see who used to be a constructor's team principal. If a new driver were to join a team, we would be able to insert it into the DRIVERS table. Similarly if a current driver were to leave, we would be able to delete it from the DRIVERS table. If a new grand prix has just finished, the result of that grand prix will update the STANDINGS table. This would be an efficient way for F1 fanatics to browse through and keep track of everything they need to know about F1.

4. Description of the application platform: (2-3 sentences)

- a. What platform will your project use (PHP/JDBC/etc.)?
- b. What is your expected application technology stack (i.e., any other things that you're using other than whether you're using PHP or JDBC)? Note that for DBMSs, we will only provide support for using the department's installation of Oracle. You are on your own for anything else.
 - i. You can change/adjust your tech stack later as you learn more about how to get started for the project via latter tutorials.

Overall, for our final application, we plan on using PHP as our coding language and Oracle as our Relational database management system (RDBMS). If possible, we would like to use React as our front-end framework.

5. An ER diagram for the database that your application will use.

