University of British Columbia, Department of Computer Science

CPSC 304

Cover Page for Project Proposal

Date: 2018-09-28

Group Member:

NAME:	STUDENT ID:	CS ID:	TUTORIAL SECTION:	EMAIL:
KYO TANG	35163104	u0q0b	T1D	kyo.hideki.tang@gmail.com
CHRISTPHER YAO	20924163	d1e1b	T1D	yaowongzhou@gmail.com
RUI ZHANG	28834166	n3v0b	T1D	zhangrui_ca@sina.com
YANYUN BU	42828146	b6h1b	T1C	iamclaudebu@gmail.com

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 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

Brief Description

The domain of our application that we're going to model is a self-driving research platform, where we're focus on store the self-driving test setting and the self-driving test record for Self-driving Car Research.

The database will mainly model the information of 4 major parts in our application: Car, Self-Driving software, Path with its geometry, and Self-Driving Test Record and Setting. User can use this application to design, execute and modify the self-driving test and collect the information from test record.

Database Specification

The database contains:

- **developer account information**
  The developer's account information for authorization to enter the application
- **test driver information**
  The information for the driver inside the car. He/She can be one of developers
- **self-driving test record and setting**
  Used to store the test setting and status for developer. It also store the test record after the testing
- **research location**
  The location for the Self-Driving Test
- **path**
  The path design created by develop. It is the part of test setting.
- **Geometry**
  The concrete point that the path need to achieve. It is represented by latitude and longitude.
- **path condition with weather and climate**
  The actual path condition information for the path. It contains weather type, the road type(road or highway etc), climate and season, and the day or the night. It can be updated by developer.
- **Self-Driving software information**
  The software information contains software's version information and update time. It is updated by developer.
- **Self-Driving software console log**
  Used to store the console log information from self-driving software.
- **car information**
  The car information for self-driving test. It can be add or delete by developer.
- **Department**
  The department information for developers, software, and test driver

Above entities contains information about the self-driving software research database. The redundant detail information has been removed. The more concrete information will be provided by in the further part.

The main users for our application are the developer and test driver.
The developer can directly create a test for self-driver research and the driver can directly modify the test setting on the car.
For example, developers(user) can log into the application for test setting and test record extraction. In the test setting part, developers(user) are able to create a new daily test and fill in the test information like car type and path based on today's weather and the road of the path. In addition, developers(user) also can create new path and new path condition for testing. After the setting, the application will store this test setting for further testing. This test will be automatically or manually started by test drivers(user). Test Driver can execute and modify the test setting when the test is proceeding. The modification will be updated on the database. After the test, a develop can search the test record including the software's console log by given filter, such as selecting the test in the raining. Developers can use the test record, for example, the brake stopping distance, for software improvement.

With the sufficient relational record, our database will provide user a powerful functionality of searching for the testing information. Based on some search option and one keywords in the car type, path, path condition, software version etc., user can directly find all corresponding data and related data on the application through our database.

Application Platform

We will build our project using CS department's oracle database as our RDBMS. The programming language that we decide to use is PHP. We do not use any other technology at this step.

We will try our best to follow our plan in this proposal in the development of our project. However, some modification in further development may be conducted if necessary.