**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

**Schemas:**

The functional dependencies in all table will contains primary keys. The LHS for all functional dependencies are keys. Thus, all tables involving functional dependencies are 3NF or BCNF.

**1.     Developer**

Definition:

Developer (accountNo:integer, password:string, name:string)

Primary Key: accountNo

                    Constraints: None

        FDs:

                    accountNo->password

                    accountNo->name

**2.     Test Driver**

Definition:

TestDriver (accountNo:integer, driverid:string, phoneNo:integer)

Primary Key: (accountNo,driverid)

Foreign Key: accountNo references Developer

Constraints: Weak Entity. Each Test Driver must be a developer.

        FDs:

                    accountNo,driverlicense-> phoneNo

**3.     Car**

Definition:

Car (carid:integer, cartype:string, deviceid:integer)

Primary Key: carid

Foreign Key: deviceid references SelfDrivingDevice

Constraints: The relationship between Car and Self Driver Derive is one-to-one.

        FDs:

                    carid->cartype

carid->deviceid

**4.     Self-Driving Device**

Definition:

SelfDrivingDevice (deviceid:integer, deviceVersion:string, carid:integer)

Primary Key: deviceid

Foreign Key: carid references Car

Constraints: The relationship between Car and Self Driver Derive is one-to-one.

FDs:

deviceid->deviceVersion

deviceid->carid

**5.     Self-Driving Software**

Definition:

SelfDrivingSoftware (versionid:integer, updatetime:string, comment:string)

Primary Key: versionid

FDs:

versionid->updatetime

versionid->comment

**6.     The list of Software in  Car**

Definition:

ListSoftwareInCar (carid:integer, versionid:integer)

Primary Key: (carid, versionid)

Foreign Key:

carid references Car

versionid references SelfDrivingSoftware

**7.     Self-Driving Software Record**

Definition:

SelfDrivingSoftwareRecord (swrecordid:integer, consolelog:string, versionid:integer)

Primary Key: swrecordid

Foreign Key: versioned references SelfDrivingSoftware

Constraints: The relationship between Self-Driving Software Record and Self-Driving Software is many-to-one

FDs:

swrecordid->consolelog

swrecordid->versionid

**8.     Self-Driving Test Record And Setting**

Definition:

SelfDrivingTestRecordAndSetting (recordid:integer, status:string, carid:integer, versionid:integer, swrecordid:integer, pathid:integer, driverid:integer, fromdatetime:string, todatetime:string)

Primary Key: recordid

Foreign Key:

carid references Car

versionid references SelfDrivingSoftware

swrecordid references SelfDrivingSoftwareRecord

pathid references Path

driverid references TestDriver

FDs:

recordid->status

recordid->carid

recordid->versionid

recordid->swrecordid

recordid->pathid

recordid->driverid

recordid->fromtime

recordid->totime

**9.     Path**

Definition:

Path (pathid:integer, city:string, location:string, startpoint:string, endpoint: string, pathcondid:integer)

Primary Key: pathid

Foreign Key:

Pathcondid references PathCondition

Constraints: The relationship between Path and Path Condition is many-to-one

FDs:

pathid->city

pathid->location

pathid->startpoint

pathid->endpoint

pathid->pathcondid

**10. Path Condition**

Definition:

PathCondition (pathcondid:integer, roadtype:string, weather:string, climate:string, dayornight:string)

Primary Key: pathcondid

FDs:

pathcondid->roadtype

pathcondid->weather

pathcondid->climate

pathcondid->dayornight

**11. Geometry**

Definition:

Geometry (geoid:integer, lat:integer, lon:integer, pathid:integer)

Primary Key: (geoid, pathid)

Foreign Key: pathid

Constraints: Weak Entity. Each Geometry must be related with one Path.

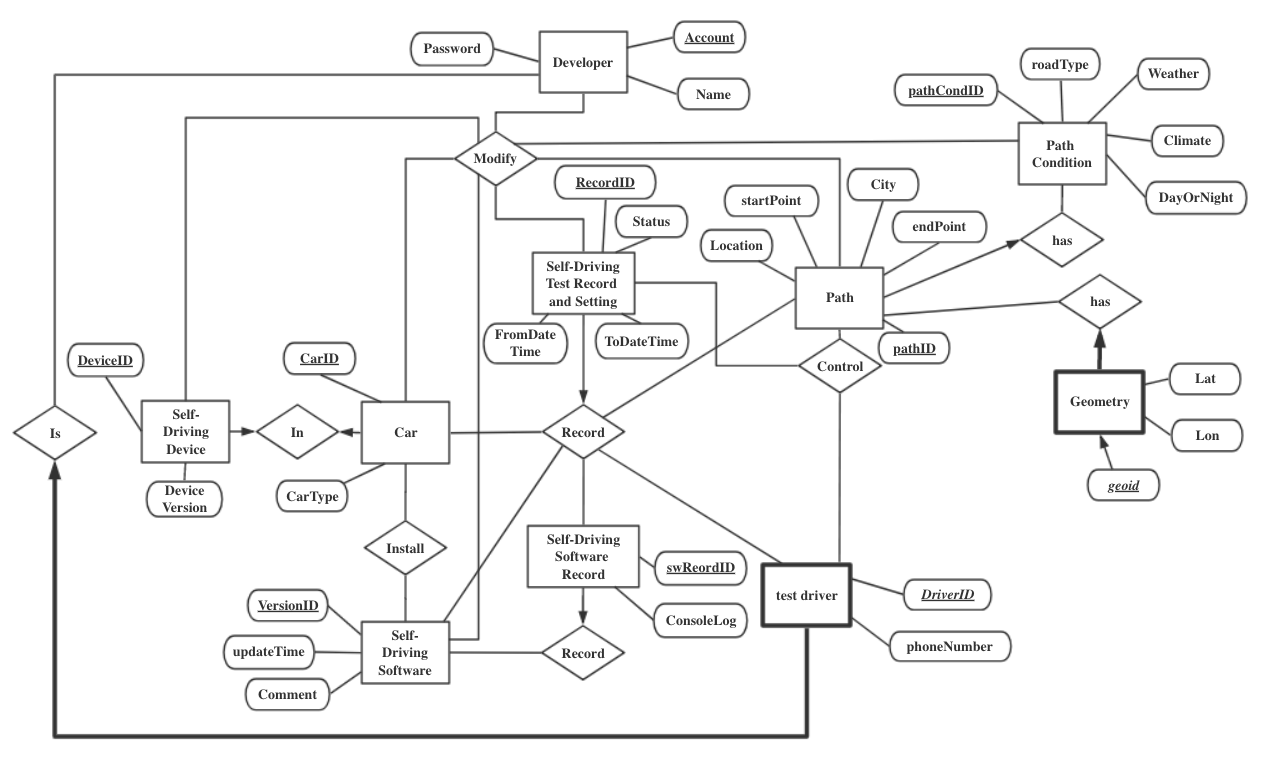
FDs:

geoid, pathid->lat

geoid, pathid->lon

geoid, pathid->pathid

**ER-diagram:**



Note:

1.The italic and bold attributes are the key of a weak entiry.

2.Developer can create and modify Car, Self-Driving Device, Self-Driving Software, Self-Driving Software Record, Path, and Path Condition.

3.The geometry for each Path is generated by the other GPS software in our application depending on the startpoint and endpoint.

4.The Self-Driving Software Record is generated by Self-Driving Software.

5. Car must contains at least one Self-Driving Software

6. Self-Driving Software may not be installed in any Car.

7. Self-Driving Test Record and Setting must has information for Car, Self-Driving Software, Path, and Test Driver.

8. Self-Driving Test Record and Setting may not contain Self-Driving Software Record. It depends on status. If the status is “finished”, Self-Driving Test Record and Setting must contain Self-Driving Software Record.

**SQL DDL：**

1.     Developer

CREATE TABLE Developer

(accountNo INT PRIMARY KEY,

password INT,

name CHAR(30))

2.     Test Driver

CREATE TABLE TestDriver

(accountNo INT,

driverID CHAR(30),

phoneNo INT,

PRIMARY KEY (accountNO, driverID)

accountNo REFERENCES Developer

ON DELETE CASCADE

ON UPDATE CASCADE)

3.     Car

CREATE TABLE Car

(carID INT PRIMARY KEY,

cartype CHAR(30),

deviceID INT,

deviceID REFERENCES SelfDrivingDevice

ON DELETE SET DEFAULT

ON UPDATE CASCADE)

4.     Self-Driving Device

CREATE TABLE SelfDrivingDevice,

(deviceID INT PRIMARY KEY,

deviceVersion CHAR(30),

carID INT,

carID REFERENCES Car

ON DELETE SET DEFAULT

ON UPDATE CASCADE)

5.     Self-Driving Software

CREATE TABLE SelfDrivingSoftware

(versionID INT PRIMARY KEY,

updatetime CHAR(30),

comment CHAR(1000))

6.     The list of Software in Car

CREATE TABLE ListSoftwareInCar

(carID INT,

versionID INT,

PRIMARY KEY (carID, versionID),

carID REFERENCES Car

ON DELETE CASCADE

ON UPDATE CASCADE,

versionID REFERENCES SelfDrivingSoftware

ON DELETE CASCADE

ON UPDATE CASCADE)

7.     Self-Driving Software Record

CREATE TABLE SelfDrivingSoftwareRecord

(swrecordID INT PRIMARY KEY,

consolelog CHAR(1000),

versionID INT,

versionID REFERENCES SelfDrivingSoftware

ON DELETE SET DEFAULT

ON UPDATE CASCADE)

8.     Self-Driving Test Record And Setting

CREATE TABLE SelfDrivingTest

(recordID INT PRIMARY KEY,

status CHAR(30),

carID INT,

versionID INT,

swrecordID INT,

pathID INT,

driverID INT,

fromdatetime CHAR(30),

todatetime CAHR(30),

carID REFERENCES Car

ON DELETE SET DEFAULT

ON UPDATE CASCADE,

versionID REFERENCES SelfDrivingSoftware

ON DELETE SET DEFAULT

ON UPDATE CASCADE,

swrecordID REFERENCES SelfDrivingSoftwareRecord

ON DELETE SET DEFAULT

ON UPDATE CASCADE,

pathID REFERENCES Path

ON DELETE SET DEFAULT

ON UPDATE CASCADE,

driverID REFERENCES TestDriver

ON DELETE SET DEFAULT

ON UPDATE CASCADE)

9.     Path

CREATE TABLE Path

(pathID INT PRIMARY KEY,

city CHAR(30),

location CHAR(30),

startpoint CHAR(30),

endpoint CHAR(30),

pathcondID INT,

pathcondID REFERENCES PathCondition

ON DELETE SET DEFAULT

ON UPDATE CASCADE)

10.   Path Condtion

CREATE TABLE PathCondition

(pathcondID INT PRIMARY KEY,

roadtype CHAR(30),

weather CHAR(30),

climate CHAR(30),

dayornight CHAR(30))

11.   Geometry

CREATE TABLE Geometry

(geoID INT,

lat INT,

lon INT,

pathID INT,

PRIMARY KEY (geoID, pathID),

pathID REFERENCES Path

ON DELETE CASCADE

ON UPDATE CASCADE)