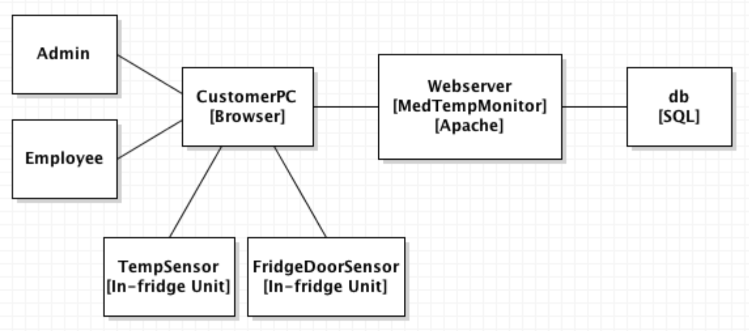
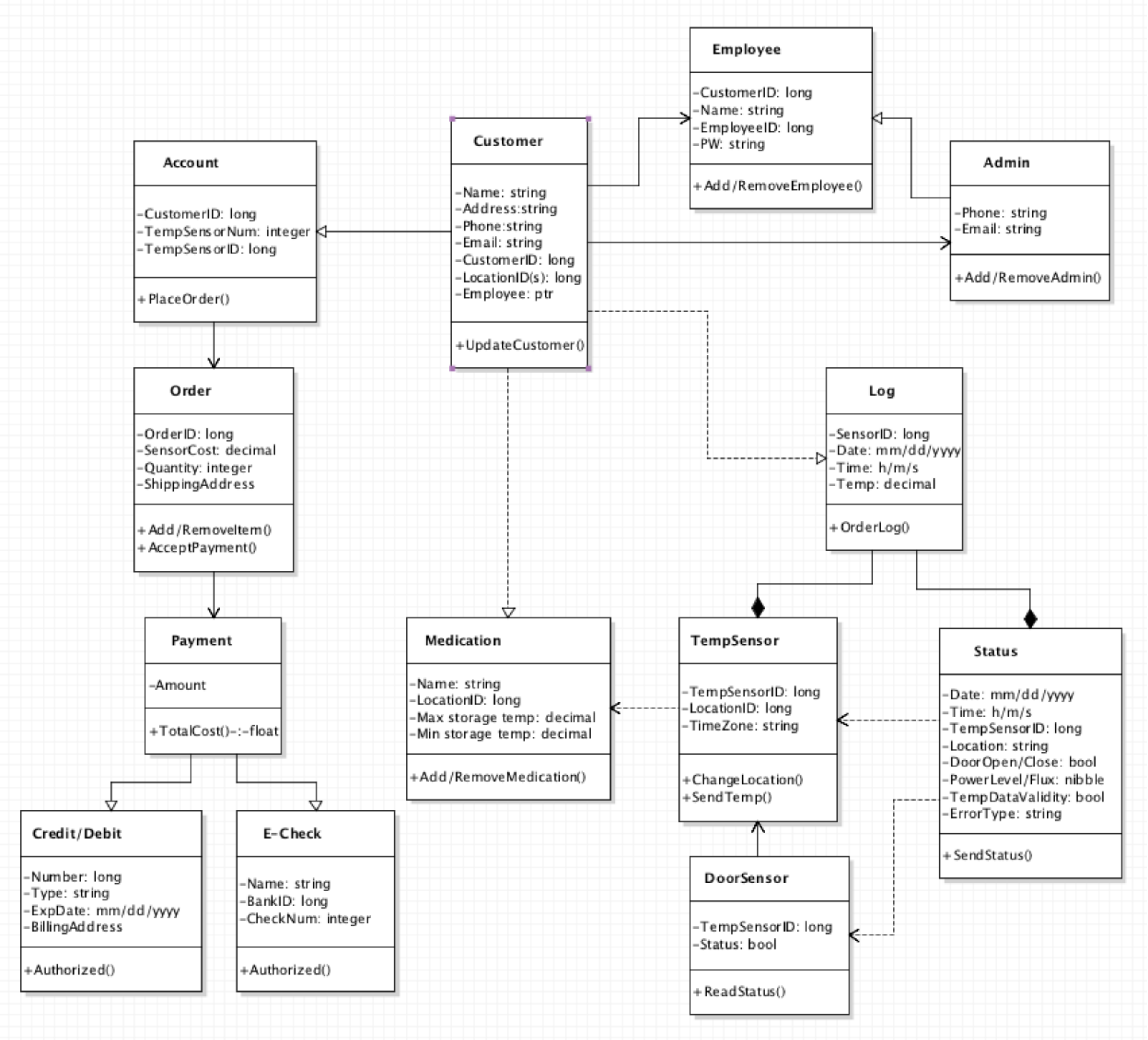
ECE 411, fall 2016
Professor Faust
T07: Jessica Blasch, Tyson Gieszler, JP Grattan, Charley Hill
Assignment: hw 5
Due: 17 November, 2016

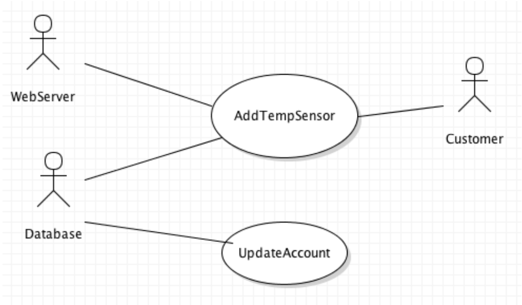
1. Draw a UML physical view of the entire system (temperature sensors, server, db, users) with as much detail and specificity as you can.


Admin
Employee
CustomerPC [Browser]
Webserver [MedTempMonitor] [Apache]
db [SQL]
TempSensor [In-fridge Unit]
FridgeDoorSensor [In-fridge Unit]

2. Write a UML class diagram to describe the data that resides on the server.

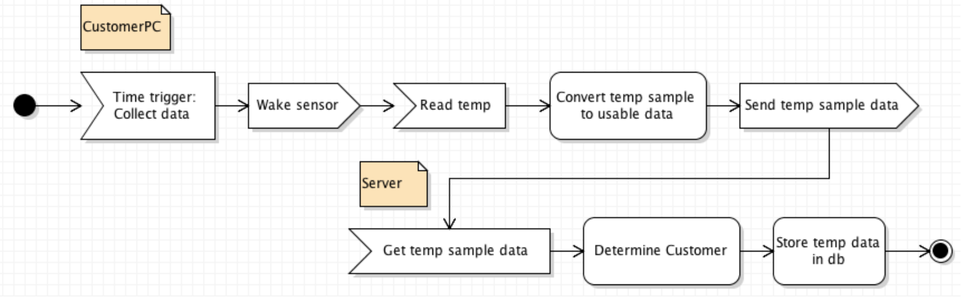


3. Write a UML case description for “Adding a Vaccine Temperature Sensor” to describe the action of adding a new temperature sensor. It should be detailed and comprehensive.



Use Case	Adding a Vaccine Temperature Sensor
Actors	Customer, Database, Webserver
Description	This use-case occurs when a customer submits a request to add a temperature sensor via the Webserver. The Webserver will prompt the customer for their user log-in information. If they do not have an account, they will be prompted to set up an account and their information will be stored in the database as a new entry. They will then be able to order a new temperature sensor or register a pre-existing one as part of the registration process.
Stimulus	Customer registers temperature sensor
Response	Verify user is valid customer with valid temperature sensor registration and that the sensor is not registered elsewhere. If valid trigger UpdateAccount.

4. Write a detailed UML activity or integration view for the temperature sensor to periodically wake up, sense temperature and send to the database. Include any actions that need to take place on the server/database side.



5. Write the specification for the HTTP GET string that will be used by the vaccine temperature monitor and the database to communicate temperature samples. Be brief, but unambiguous. It should be clear enough that two individuals, one working on the server side (writing code to parse the string and put the appropriate values in the database) and another working on the temperature monitor code (responsible for the code to output the HTTP GET URL string) can do their work independently and have confidence that, when integrated, their code will work together.

```
GET request-URI?locationID="String <= 15 characters
ASCII"&temp="TTT.T"&timeDate="YYYYMMDDZHRMNSC"&status="####" HTTP/1.1
Connection: Close // Want to make sure connection closes
(blank line)
/*
Location notes
  15 characters
  case sensitive
  alphanumeric
temp notes
  Fahrenheit
timeDate Notes
  YYYY  = year - numerical
  MM = month - numerical
  DD = day - numerical
  Z = International time zone - alphabet lower case
  HR = Hour - numerical
  MN = minute - numerical
  SC = second - numerical
status
  Four digit integer
  Not currently used
*/
```